

Jonathan Inglis

**BEGINNERS' MICRO GUIDES**

# Amstrad CPC464









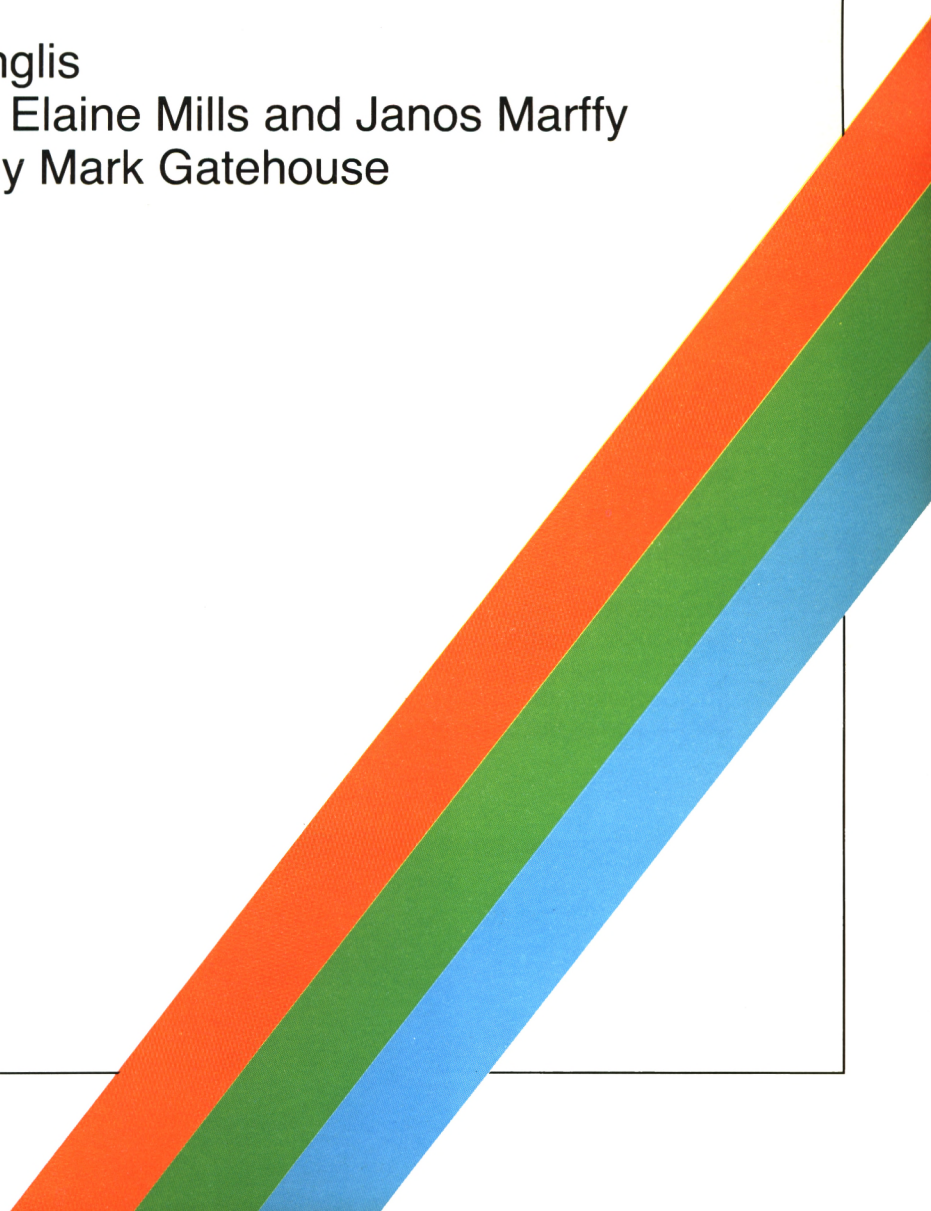
BEGINNERS' MICRO GUIDE

# Amstrad CPC464

By Jonathan Inglis  
Illustrations by Elaine Mills and Janos Marffy  
Photography by Mark Gatehouse



**GRANADA**





Published by Granada Publishing 1985  
Granada Publishing  
8 Grafton Street, London W1X 3LA

Copyright © Jonathan Inglis 1985  
Illustrations copyright © Granada Publishing 1985

*British Library Cataloguing in Publication Data*  
Inglis, Jonathan

Amstrad CPC464. – (Beginners' micro guides 6)

1. Amstrad CPC464 (Computer)

I. Title II. Series

001.64'04 QA76.8.A4

ISBN 0-246-12460-1

Printed in Italy by  
New Interlitho, Milan

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of the publishers.



# Contents

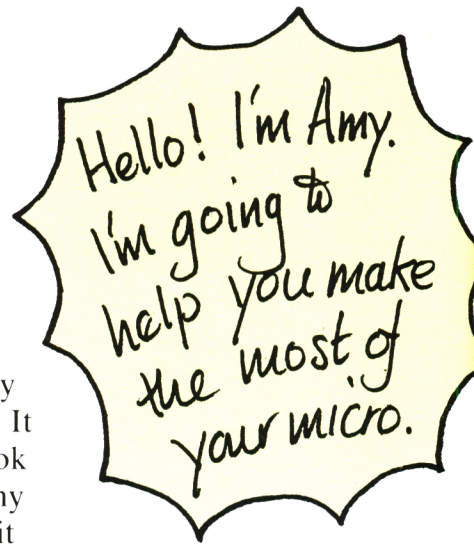
The Amstrad CPC464 .....	4
The Keyboard .....	8
Getting Started .....	10
Writing Programs .....	15
Planning Screen Displays .....	19
Inside Your Computer .....	27
Variables .....	29
Loops .....	34
Sound Effects .....	40
Drawing .....	42
Loading and Saving .....	49
Putting in Information .....	54
Making Decisions .....	58
Index .....	64



# The Amstrad CPC464

The Amstrad CPC464 is an exciting machine. You can play games on it, or use it like a notebook to store information. It can draw multi-coloured patterns and play music. This book will teach you how to make your Amstrad CPC464 do many different jobs. There is a lot to learn; at first you may find it slow going, but don't be in too much of a hurry. A computer is a complicated machine. It takes skill and patience to use it properly.

It also needs care. The computer is delicate so treat it carefully. Don't hammer the keys and don't be tempted to take the lid off just to look inside! Later in this book we'll show you what's inside.





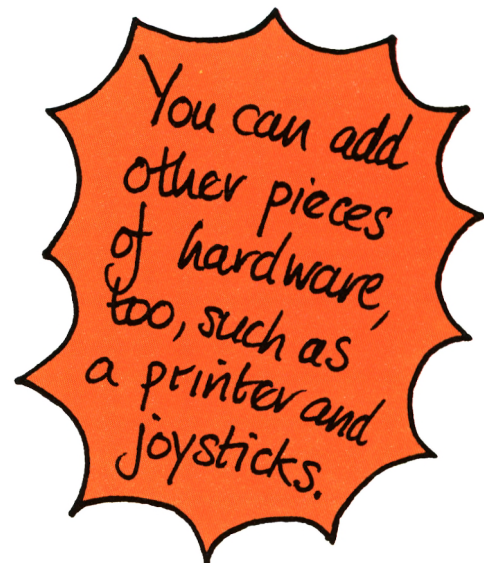


### The computer system

The equipment in a computer system is called **HARDWARE**.

The picture shows two pieces of hardware:

1. The **MICROCOMPUTER** itself, which is usually just called the **COMPUTER**. The Amstrad CPC464 also has a built-in **CASSETTE RECORDER**, which can store any programs you may want to keep.
2. A special type of television called a **MONITOR**. This can also be called a **VDU** or **VISUAL DISPLAY UNIT**.



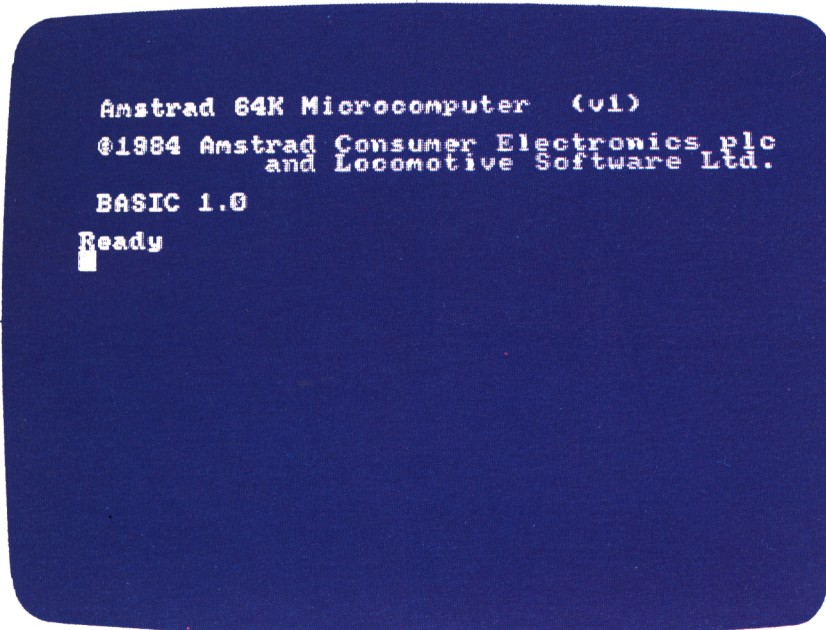


# Setting up

Ask an adult to put a plug on the power lead for the monitor (the television). At the front of the monitor are two leads. Connect the smaller plug to the socket marked 5V DC, at the back of the computer. Connect the larger plug to the socket marked MONITOR.



Now connect the other plug on the monitor to the mains, and switch on. On the front of the monitor is a switch marked POWER. Switch this on too. On the right hand side of the computer is a switch marked POWER; switch this to the ON position. The red light on the computer should come on, and you should see a message on the monitor screen.



Adjust the 'brightness' control on the right hand side of the monitor, so that the message is clear and sharp. Remember, you may spend a lot of time staring at the screen and it isn't good for your eyes if the picture is too bright.

## \*MORFAX\*

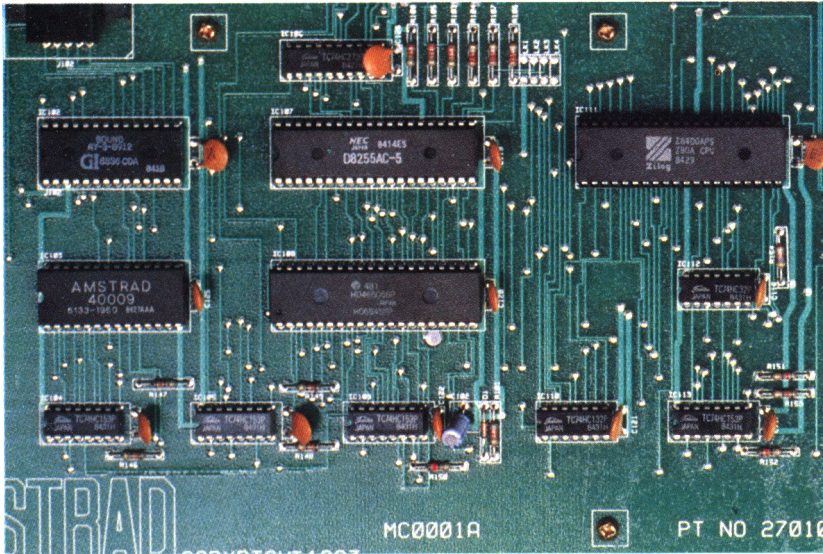
The message on the screen tells you what version of BASIC your Amstrad CPC464 uses (BASIC 1.0). The word 'Ready' tells you that the computer is ready for you to start giving instructions.



# What is a computer?

Your computer is more than just an electronic typewriter. It works by obeying sets of instructions. These can be written in BASIC, a special language which the computer obeys but which is also easy enough for beginners to understand.

A set of BASIC instructions is called a PROGRAM. Putting programs into your computer is called PROGRAMMING.



Inside your computer are all sorts of microchips and electronic circuits. Some of these translate the instructions you give into coded electronic signals. Each word or instruction in a program has its own code. These are stored in chips which form the computer's MEMORY. Once a program is in the memory, it can be used over and over again.

When you first switch on your computer, it cannot yet do very much. It needs programming. You can buy ready-made programs on tape, and feed them in using the cassette recorder. These tapes and the programs on them are known as SOFTWARE.

Using ready-made software is fun, but not as interesting as writing your own programs. To do this you type the programs in directly, using the keyboard of your computer.

```
10 LET Z=2.25
20 LET Y=7.575
30 PRINT Z
40 PRINT Y
50 PRINT Z+Y
```

**A computer program  
is a list of instructions  
placed in numbered lines.**



**\*MORFAX\***

You can buy ready-made programs in several different forms. The most common are on cassette tapes, but if you have a disk drive you can buy software on floppy disks too.





# The Keyboard

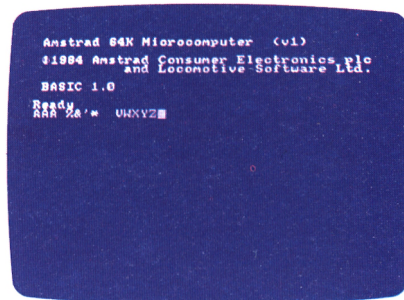
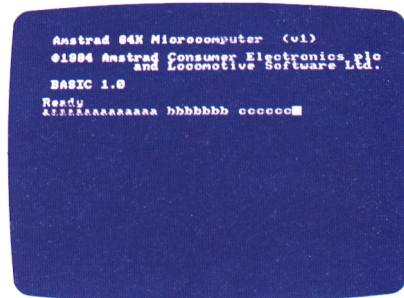
Your Amstrad CPC464 has five rows of keys, a long key at the bottom and another group of keys on the right. All of these keys together are known as the **KEYBOARD**.

Now look at the screen. On it you will see a small block. This is known as the **CURSOR**. Anything you type in will be printed to the left of this cursor. Press one of the letter or number keys. That letter or number will be printed on the screen. If you keep your finger on the key, it will be printed again and again until you take your finger off.

Type a few letters. If you want to leave a space between two letters press the **SPACE BAR** (the long key at the bottom of the keyboard).

## Rubbing out

At the top right hand corner of the keyboard is a green key marked **DEL**. When you press this it rubs out or 'deletes' the letter to the left of the cursor. Keep your finger on this key and it rubs out everything you have typed in, though you cannot delete the cursor.





# Keyboard control

## The SHIFT keys

Just above the bottom row of the keyboard are two green keys marked SHIFT. Press one of these, and at the same time press one of the letter keys. You will see that letter printed as a capital letter. Keep your finger on the SHIFT key. Now press any of the keys with numbers or symbols printed on. On the screen you will see the symbol on the top half of that key.



## Capital letters

If you want all your letters to be 'upper case' ie capitals, then press the green CAPS LOCK key on the left hand side of the keyboard. Most of the programs in this book use capital letters only. To switch off the CAPS LOCK, press it again.



## The numeric keypad

To the right of the main part of the keyboard you will see the 'numeric keypad'. This extra set of keys makes it easier if you need to type in a lot of numbers. Typing in the numbers from this keyboard has exactly the same effect as typing them in from the main keyboard.

## The ESCAPE key

In the top left hand corner of the keyboard is a red key marked ESC. This is known as the ESCAPE key. When you press it you will see that it prints the message \*Break\*, and the cursor moves down a line on the screen.



# Resetting the computer

If you find that your Amstrad CPC464 isn't behaving as you expect, you may need to reset it. To do this, press the CTRL, SHIFT and ESC keys at the same time. This clears the computer's memory. Always do this before loading in a game.

If your screen is beginning to look a little cluttered and you want to clear it then type in, on a separate line:

CLS

and press the ENTER key.





# Getting Started

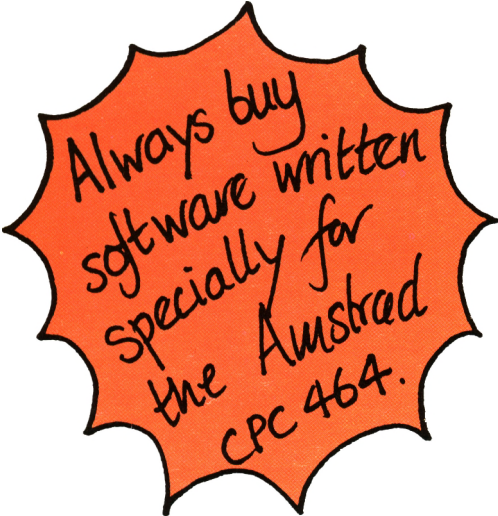
Now that you can use the keyboard, you can begin to give your Amstrad CPC464 instructions. But you can't just type in anything. What you type in must make sense to the computer.

Your Amstrad CPC464 has been programmed to recognize a language called BASIC. If you want to learn how to use your computer, you will have to learn about BASIC too.

BASIC is made up of words, numbers and symbols. You must be very careful how you write in BASIC. Computers need precise instructions and there are strict rules that you must follow, called 'syntax'. A mistake which breaks these rules is called a SYNTAX ERROR. Typing in a wrong letter or symbol can stop a program from working.



Mistakes in programs are also called bugs.



Always buy software written specially for the Amstrad CPC464.



## Basic buying

Most home computers use BASIC. Unfortunately they don't all use the same version of BASIC. You might think that a program which works on other computers will work on the Amstrad CPC464.

**BE WARNED! IT MAY NOT!**

If you are copying out a program, first make sure that it will work on the Amstrad CPC464. If you don't, you could be in for a disappointment.

If you buy tapes, make sure that they are for the Amstrad CPC464. Atari, BBC Micro, Sinclair Spectrum or Commodore 64 software will not work. Software can only work on the computers for which it has been written.



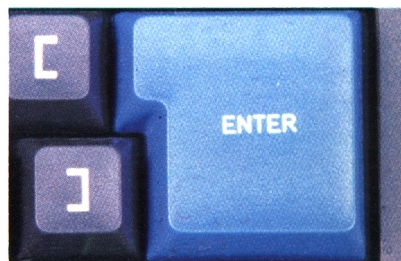
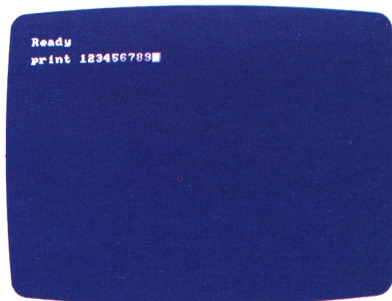
# PRINT

BASIC uses many words that you will recognize. The first BASIC word to look at is PRINT. Type in, leaving a space after the word PRINT:

```
PRINT 123456789
```

This instructs your Amstrad CPC464 to print the number 123456789 on the screen. Of course you can already see that number on your screen now. What you want is to see your Amstrad CPC464 printing it.

To do this you use ENTER, the big blue key on the right hand side of the keyboard. Press this now and you will see the number 123456789 printed on the screen.

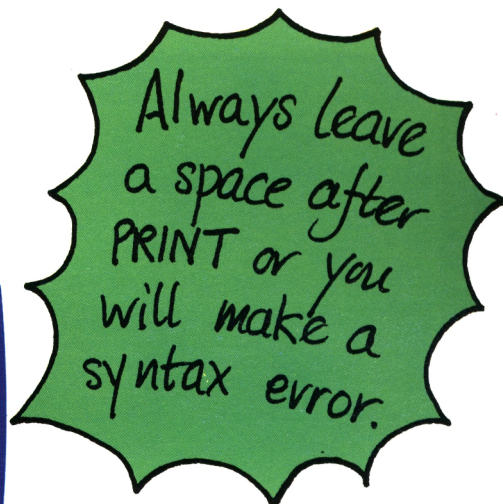
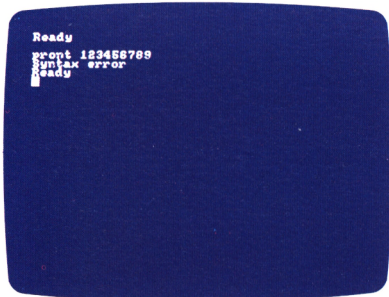


## Report messages

If your Amstrad CPC464 has carried out its instructions successfully, you will see the word READY on your screen. This is known as a REPORT.

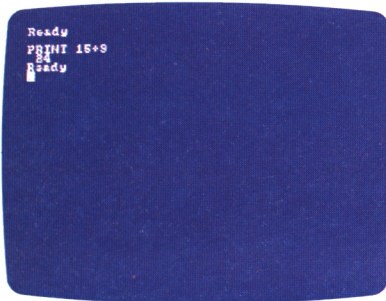
There are all sorts of report messages. They can tell you whether your Amstrad CPC464 has understood the instructions you have given it. Problems may be caused by simple errors in typing. Type in the word PRONT instead of PRINT, and you will be told that you have made a syntax error.

Try using PRINT to print some more numbers on the screen. Remember to press the ENTER key every time you want your Amstrad CPC464 to carry out its instructions.





# Symbols and sums

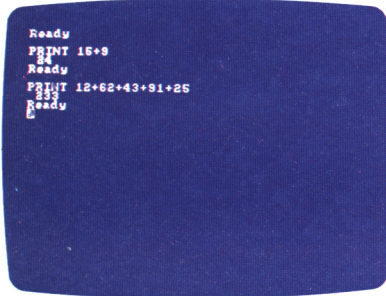


## Adding and subtracting

You can also PRINT numbers and mathematical symbols. Some of these you will know already. For instance, type this in:

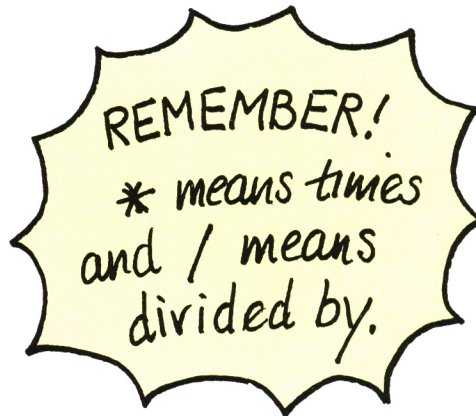
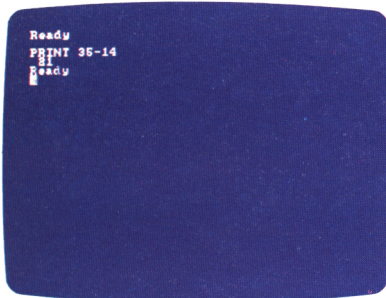
```
PRINT 15+9
```

Press the ENTER key and, hey presto! Your Amstrad CPC464 prints the answer to your sum. (To get the plus sign, press SHIFT and the key on which + is printed.)



Try this with some other numbers. You can add up more than two numbers. Remember to press the ENTER key when you want to see the answer.

You can also subtract, using the minus sign. You may use the minus and plus signs together on one line. The minus sign is on the same key as the equals sign (top row of the keyboard).



## Multiplying and dividing

In BASIC you do not use the  $\times$  or  $\div$  signs. Instead you use the \* sign for multiplication, and for division you use the / sign. The / sign is on the same key as the question mark.

To see how these work, try typing in:

```
PRINT 14*3
```

Find out the answer and then type in:

```
PRINT 90/5
```





### Decimal points

You can use decimal numbers with your Amstrad CPC464. Use the full stop sign for the decimal point. Type in:

```
PRINT 8.9*3.752
```

and press ENTER to see the answer.

### Big numbers

Have you noticed what happens when you try to use very large numbers? See what happens when you type in:

```
PRINT 987324567765
```

and press ENTER.

The Amstrad CPC464 cannot store very large numbers or fractions in their usual form. So it shortens them or expresses them in a different way.

### Separating sums

You may put more than one sum on one line, if you separate each one with a comma like this:

```
PRINT 18+30,96-50,7.2*3.9
```

Press ENTER and you will see the answer to all three sums displayed on the screen.







## Letter strings

Words and numbers can be printed on the screen using quotation marks. Any information printed in this way is called a STRING, because several characters are strung together inside the quotation marks.

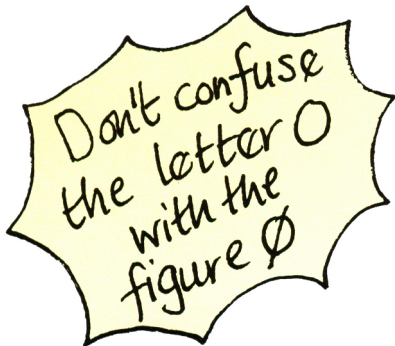
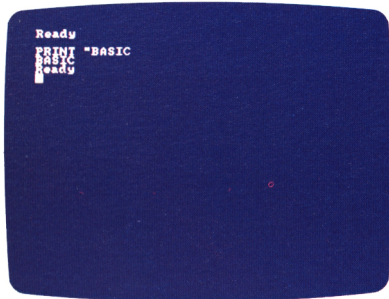
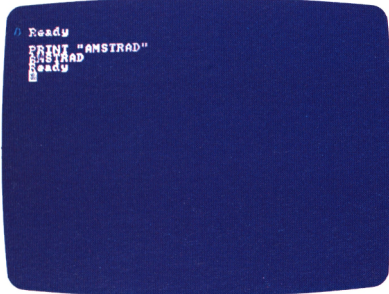
If you want to use your Amstrad CPC464 to print words, you have to use quotation marks (press SHIFT and key 2). Type in:

```
PRINT "AMSTRAD"
```

Press ENTER, and your Amstrad CPC464 will carry out its instructions. Now type in:

```
PRINT "BASIC"
```

and press ENTER. Leaving out the second set of quote marks makes no difference. Some computers do not let you do this. In some cases like this the Amstrad CPC464 will accept PRINT with quote marks only before the words you want to print, but sometimes it will require a set of quote marks after them as well.



## Number strings

You can print numbers inside quotation marks too. When you print numbers and symbols inside quotation marks your Amstrad CPC464 will not treat them as a sum. It won't do any sums inside quotation marks.

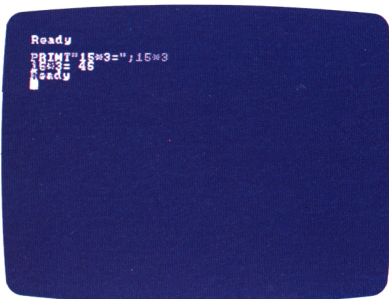
This is useful when you want to print both a sum and its answer, instead of just the answer on its own.

For example, type in:

```
PRINT "15*3="; 15*3
```

Press ENTER and what do you see? Your Amstrad CPC464 has printed the sum and its answer.

The semi-colon ; symbol is very useful. Use it when you want certain items of information to be printed next to each other.





# Writing Programs

So far you've looked at how to give your Amstrad CPC464 simple commands which it has carried out directly. But you can also make it carry out whole lists of instructions, which it is able to store in its memory. To make it do this you need to write a PROGRAM. (This American spelling, not *programme*, is always used in computing.)

Look at Amy's instructions. They are a type of PROGRAM, not for a computer but for you. They describe how to make a cup of tea.



SEE IF KETTLE HAS BOILED  
ADD MILK  
PUT KETTLE ON  
PUT TEA INTO POT  
POUR BOILING WATER INTO POT  
GET TEA CUP  
POUR WATER INTO KETTLE

Each line is an instruction, but somehow they have got jumbled up. How would you sort them out into the right order?

If you get the instructions in the right order, you can make a cup of tea. But if you get them in the wrong order you may have to go thirsty!

Of course you already know how to make a cup of tea. You know that you need to use boiling water and that you need to get a cup ready. Computers aren't so clever. To program them you need to give clear and precise instructions, in the right order. If you don't, your computer will be in as much of a mess as you would be if you tried to make a cup of tea without putting any water in the kettle first!

1. POUR WATER INTO KETTLE
2. PUT KETTLE ON
3. SEE IF KETTLE HAS BOILED
4. PUT TEA INTO POT
5. POUR BOILING WATER INTO POT
6. GET TEA CUP
7. ADD MILK
8. POUR TEA INTO CUP



# Line numbers

A program is a list of instructions written in BASIC. So that the computer knows in what order to carry out the instructions, each line of instructions is given a number. Before typing in any instruction you must put a line number. This is so that that line can be stored in the computer's memory.

When it carries out the program you've typed in, your Amstrad CPC464 looks for the lowest line number and starts there.

## Your first program

Let's see how a simple program works. Type in:

```
10 PRINT "HELLO AND WELCOME"
```

When you've done this press ENTER and type in:

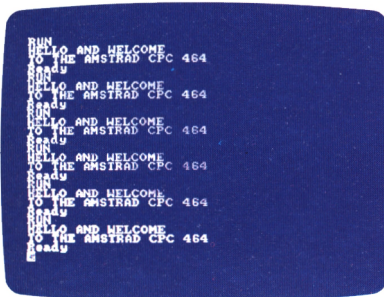
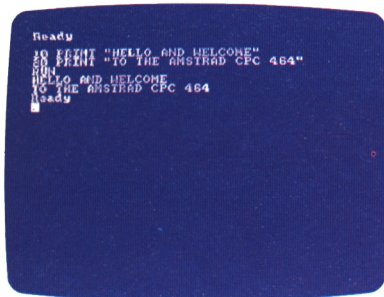
```
20 PRINT "TO THE AMSTRAD CPC464"
```

Press ENTER again. You've just written your first computer program. Congratulations! Now you need to tell your Amstrad CPC464 to carry out the program.

## RUN

Type in the BASIC word RUN and press ENTER. You should now see your message on the screen. Your computer has carried out the instructions in your program.

Once you have written a program, you can RUN it as often as you like. There's no need to type the program in again. Try running your program a few times. When you reach the bottom of the screen, the screen display moves up and the program disappears from the screen. This is known as SCROLLING.





# Listing your program

The list of instructions that make up your program is called a **PROGRAM LISTING**. If you run the program often enough, the listing scrolls off the top of the screen. Don't worry! It isn't lost. It is still stored in your Amstrad CPC464's memory.

To get the listing back, use the word **LIST**. Type in the word **LIST** now and press **ENTER**. Your listing will reappear on the screen. If you want to clear the screen first, type in the instruction **CLS** on a separate line and press **ENTER**.

## Leaving space

Look at those line numbers again. Why is there a gap of ten between each one? Why not just start at line 1 and go on to line 2?

Well, nobody's perfect. You may want to put some more instructions in between your first and second lines. A gap of ten leaves enough room for you to do this. Try typing in a new line like this:

```
15 PRINT "TO THIS INTRODUCTION"
```

Don't forget to press **ENTER**. Now type in the instruction **LIST** and press **ENTER** again. As you can see, line 15 jumps into place between lines 10 and 20.

**RUN** the program and you will see what difference line 15 makes to your screen display.

```
RUN
HELLO AND WELCOME
10 THE AMSTRAD CPC 464
READY
HELLO AND WELCOME
10 THE AMSTRAD CPC 464
READY
HELLO AND WELCOME
10 THE AMSTRAD CPC 464
READY
HELLO AND WELCOME
10 THE AMSTRAD CPC 464
READY
HELLO AND WELCOME
10 THE AMSTRAD CPC 464
READY
15 PRINT "HELLO AND WELCOME"
16 PRINT "TO THE AMSTRAD CPC 464"
READY
```

```
READY
15 PRINT "HELLO AND WELCOME"
16 PRINT "TO THE AMSTRAD CPC 464"
17 PRINT "TO THIS INTRODUCTION"
READY
```

```
READY
15 PRINT "HELLO AND WELCOME"
16 PRINT "TO THE AMSTRAD CPC 464"
17 PRINT "TO THIS INTRODUCTION"
LIST
15 PRINT "HELLO AND WELCOME"
16 PRINT "TO THE AMSTRAD CPC 464"
17 PRINT "TO THIS INTRODUCTION"
READY
```

```
READY
10 PRINT "HELLO AND WELCOME"
15 PRINT "TO THIS INTRODUCTION"
20 PRINT "TO THE AMSTRAD CPC 464"
READY
20
```





# Chopping and changing

```
Ready
10 PRINT "HELLO AND WELCOME"
20 PRINT "TO THIS INTRODUCTION"
30 PRINT "TO THE AMSTRAD CPC 464"
Ready
20
LIST
10 PRINT "HELLO AND WELCOME"
20 PRINT "TO THIS INTRODUCTION"
Ready
```

## Removing lines

As well as putting new lines into your programs you can take lines out. List the program that you have just written. What if you want to get rid of line 20? All you have to do is type in:

20

Press ENTER. Now list the program. Line 20 has disappeared from your program listing.

```
Ready
10 PRINT "HELLO AND WELCOME"
20 PRINT "TO YOUR INTRODUCTION"
Ready
LIST
10 PRINT "HELLO AND WELCOME"
20 PRINT "TO YOUR INTRODUCTION"
Ready
```

If you want to get rid of a line in a program, just type in its line number and press ENTER. Make sure that it's the right one though!

```
Ready
10 PRINT "HELLO AND WELCOME"
20 PRINT "TO YOUR INTRODUCTION"
Ready
RENUM
10 PRINT "HELLO AND WELCOME"
20 PRINT "TO YOUR INTRODUCTION"
Ready
```

## Changing a line

If you want to change the instructions in one of your lines, write the whole line again, including the line number, and press ENTER. The new version of your line replaces the old one.

## Renumbering lines

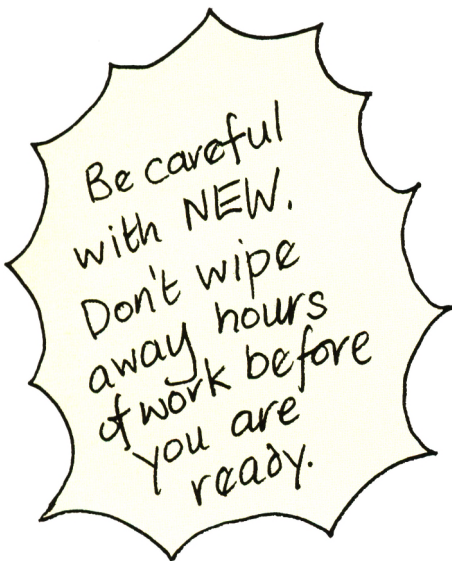
Programs always look neater with a gap of ten between each line number. To tidy up a program listing which contains different gaps, type in on a separate line:

RENUM

Press ENTER and then list the program. This renumbers the lines in your program. Now, of course, you can add some more lines in between if you want.

## NEW

Sometimes you may want to get rid of a whole program. To do this, type in the word NEW on a separate line, and press ENTER. NEW removes the program from your Amstrad CPC464's memory. Now you can start to type in a new program.





# Planning Screen Displays

You can use certain instructions in your programs to create eye-catching screen displays.

Try the program below. It uses asterisks to decorate the top and bottom of the screen display. On some lines you will see the word PRINT used on its own. Used in this way, each PRINT leaves a blank line on the screen. This spaces out the display.

```
10 CLS
20 PRINT:PRINT
30 PRINT "*****
*****"
40 PRINT:PRINT:PRINT:PRINT
50 PRINT "          SCREEN DISPLAY"
60 PRINT
70 PRINT "          BY"
80 PRINT
90 PRINT "          AMSTRAD CPC464"
100 PRINT:PRINT:PRINT:PRINT
110 PRINT "*****
*****"
```

Remember to press ENTER before going on to a new line. To print the spaces, press the space bar at the bottom of the screen. Don't worry about your typing going off the edge of the screen. Your Amstrad CPC464 automatically moves on to the next line.

RUN the program. Next, list it again and make some changes. Try experimenting with the spacing, or changing the symbol used in lines 30 and 110, to make the screen display look different. Then change lines 50, 70 and 90 to give a different message, like the birthday message in the picture.

Don't get rid of this program yet; the next few pages will show you how you can add colour to your screen displays.

**\*MORFAX\***

If you find that the word Ready spoils your screen display, why not add this line to the program?

```
120 IF INKEY$=""
THEN GOTO 120
```

Now, when you run the program, the word Ready won't appear on the screen until you press one of the letter keys. You can put this line at the end of any program; just alter the line number to fit the program.





# Choosing modes

Your Amstrad CPC464 offers you a choice of three types of screen display. These are known as MODES. So far all the programming that you have done has been in MODE 1.

To see what the last program looks like in MODE 2, type in the instruction:

`MODE 2`

and press ENTER. Now list the program. MODE 2 letters are much smaller than MODE 1 letters. Mode 2 is called an 'eighty column' mode, because each line can hold up to eighty letters or characters.

Now type in the instruction:

`MODE 0`

and press ENTER, then list the program. MODE 0 letters are much bigger than those in MODE 1 or 2. In MODE 0 you can only fit twenty letters per line. To get back to MODE 1 now, just type in the instruction MODE 1 and press ENTER.

Each of these three modes has its own special features and uses. For example, MODE 0 offers you a wide range of colours. You will probably want to use this mode if you write games programs.

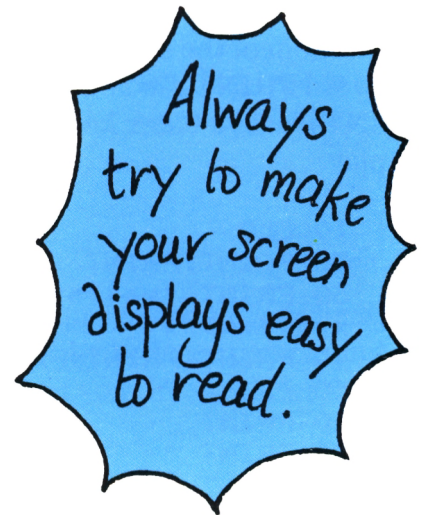
MODE 2 is more useful for business programs such as word processing, where it is important to be able to see lots of information on the screen.

## Adding colour

Screen displays look even better in colour. The Amstrad CPC464 offers you a choice of up to 26 different main colours. You cannot use all of these on the screen at the same time. In MODE 0 you can use up to sixteen at once, in MODE 1 you can use up to four and MODE 2 allows you only two.

**\*MORFAX\***

You must leave a space between the word MODE and the number following it. The instruction MODE2 will not work.





These can be used in a number of ways. You can have:

1. A coloured frame for your displays. For this you use the BASIC word BORDER.
2. A coloured background inside the border. For this you use the BASIC word PAPER.
3. Coloured letters or shapes printed on the screen. For this you use the BASIC word PEN.

First type in the instruction MODE 1, and press ENTER.

### **BORDER**

You can use any of the 26 colours listed here for your coloured borders. Type in the instruction BORDER and then any of the numbers in the chart, for example BORDER 19. Press ENTER, and you get a Sea Green border! Experiment with some of the other numbers to see what the colours look like.

### **MODE 1 colours**

In MODE 1 you have a choice of four colours for the background and letters used in your screen displays. These are:

- 0 Blue
- 1 Yellow
- 2 Cyan
- 3 Red

### **PAPER**

To change the background colour of the screen, use the word PAPER. Type in:

**PAPER 3: CLS**

and press ENTER. This gives you a red background. The CLS clears the screen to make way for the new background colour. To get back to the blue background type in PAPER 0:CLS

### **\*BORDER\* COLOURS**

- 0 Black
- 1 Blue
- 2 Bright Blue
- 3 Red
- 4 Magenta
- 5 Mauve
- 6 Bright Red
- 7 Purple
- 8 Bright Magenta
- 9 Green
- 10 Cyan
- 11 Sky Blue
- 12 Yellow
- 13 White
- 14 Pastel Blue
- 15 Orange
- 16 Pink
- 17 Pastel Magenta
- 18 Bright Green
- 19 Sea Green
- 20 Bright Cyan
- 21 Lime Green
- 22 Pastel Green
- 23 Pastel Cyan
- 24 Bright Yellow
- 25 Pastel Yellow
- 26 Bright White

### **\*MORFAX\***

Always leave a gap between the word BORDER and the number you choose. The instruction BORDER19 will not work.



## PEN

To change the colour of the letters used in the screen display, use the word PEN. Type in:

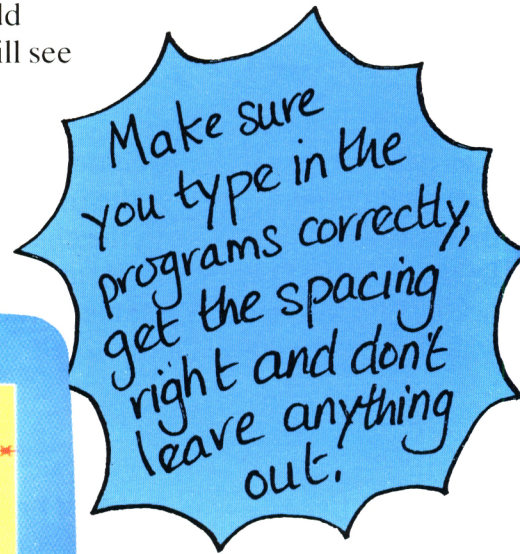
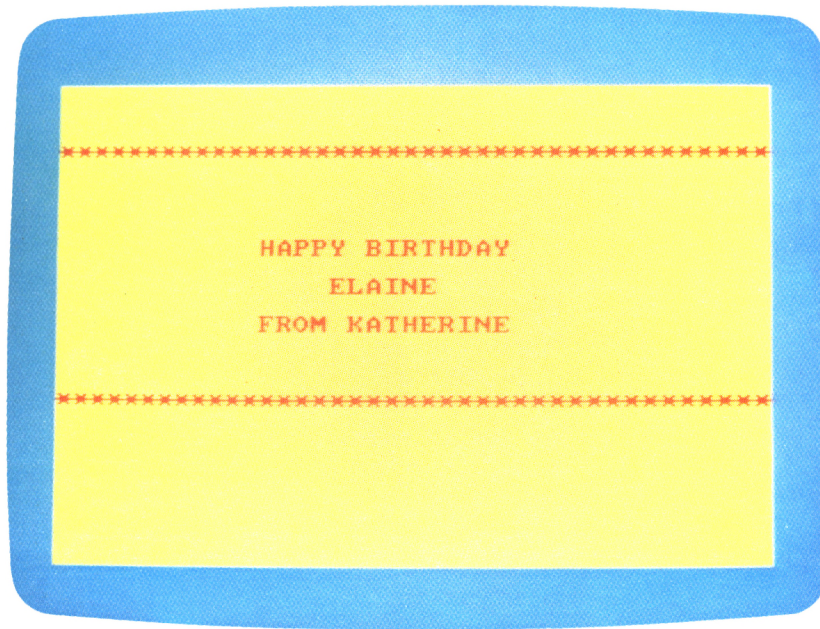
PEN 2

and press ENTER. This gives you cyan letters. To get back to yellow letters, type in PEN 1.

## Putting colour in programs

These instructions can be built in to your programs. Add these lines to the program on page 19; run it and you will see what they do.

```
10 MODE 1  
15 BORDER 14: PAPER 1:CLS: PEN 3
```



\*MORFAX\*

To get back to the normal screen colours, type in (on one separate line) the following instructions:

```
BORDER 1:PAPER 0:CLS: PEN 1
```

and press ENTER.



# Lucky Dip

Here are some programs for you to type in and try. Don't worry if you don't understand everything in them at the moment. Just have fun.

Make sure you type them in correctly. If you don't, your computer will let you know.

```
10 MODE 0
20 BORDER 0
30 PAPER 5:CLS
40 FOR N=1 TO 480
50 C=RND*14
60 PEN C
70 PAPER C+1
80 X=(RND*128)+127
90 PRINT CHR$(X);
100 NEXT N
110 IF INKEY$="" THEN 110
120 MODE 1: BORDER 1
130 PAPER 0:CLS: PEN 1
```





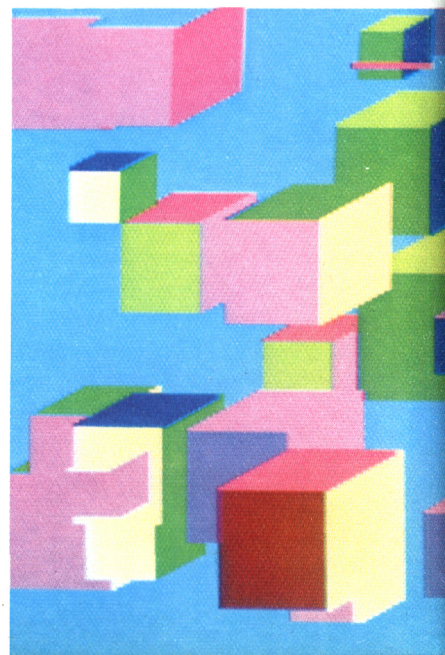
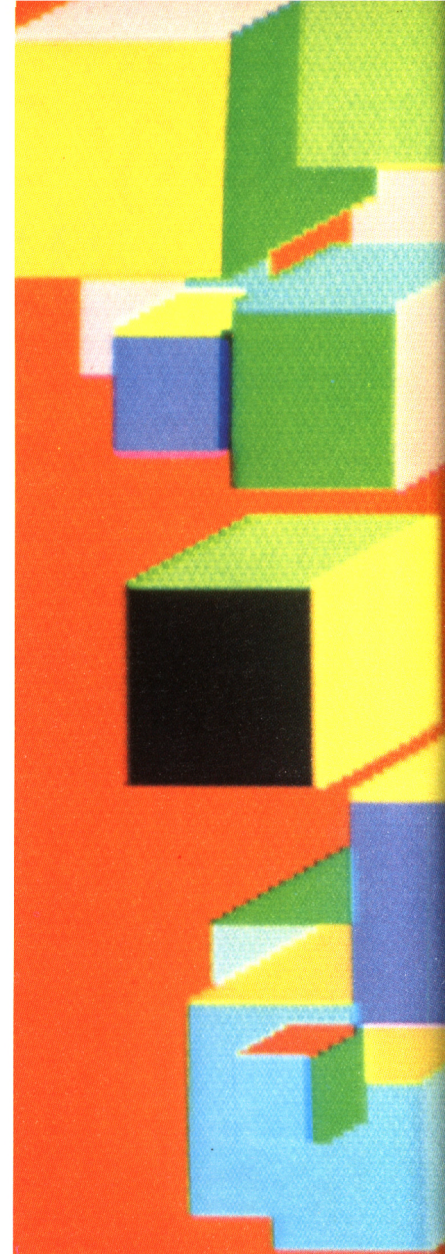
This program takes quite a long time to run. When the screen display stops changing and you want to clear the screen, press any key.

```
10 MODE 0
20 FOR N=1 TO 100
30 INK RND*15,RND*26
40 X1=INT(RND*640)
50 Y1=INT(RND*400)
60 L=INT(RND*50)+25
70 C1=INT(RND*11)+3
80 C2=C1-2:C3=C1-1
90 FOR Y=Y1 TO Y1+L STEP 2
100 PLOT X1,Y
110 DRAW X1+L,Y,C1
120 NEXT Y
130 Y=Y1+L
140 FOR T=Y TO Y+L/3
150 PLOT X1,T: DRAW X1+L,T,C2
160 DRAW X1+L,T-L,C3
170 X1=X1+2
180 NEXT T
190 NEXT N
200 IF INKEY$="" THEN 200
210 MODE 1: BORDER 1:INK 0,1: INK
1,24: CLS
```

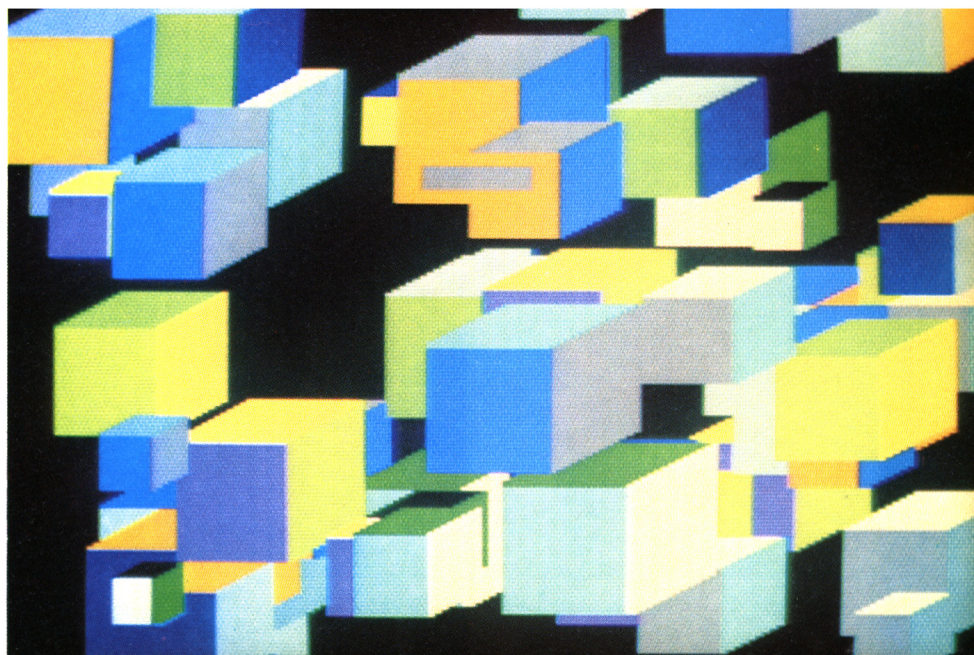
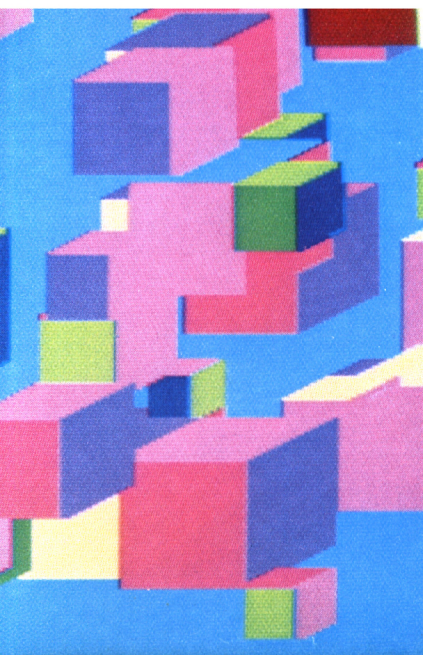
When you've run these programs a few times, try changing them. Change the colours of your borders and backgrounds first, then try changing some of the numbers. Even if this doesn't always work, you'll find out a lot about your computer.

#### \*MORFAX\*

To save time typing in line numbers, type in (on a separate line) the instruction AUTO. Now press ENTER and you will see the number 10 on the screen. Type in your line and then press ENTER again. The next line number appears automatically. When you reach the line number after the end of the program, press ESC. This switches off the automatic line numbering.



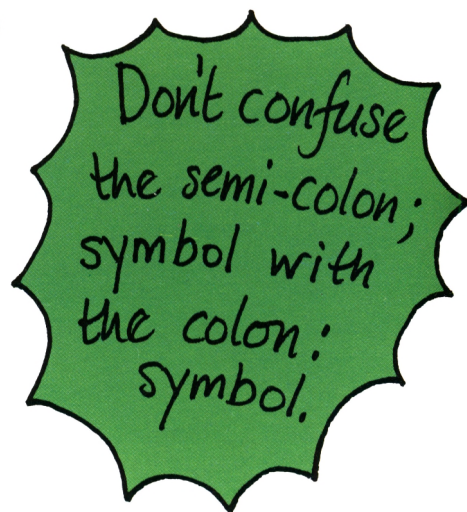




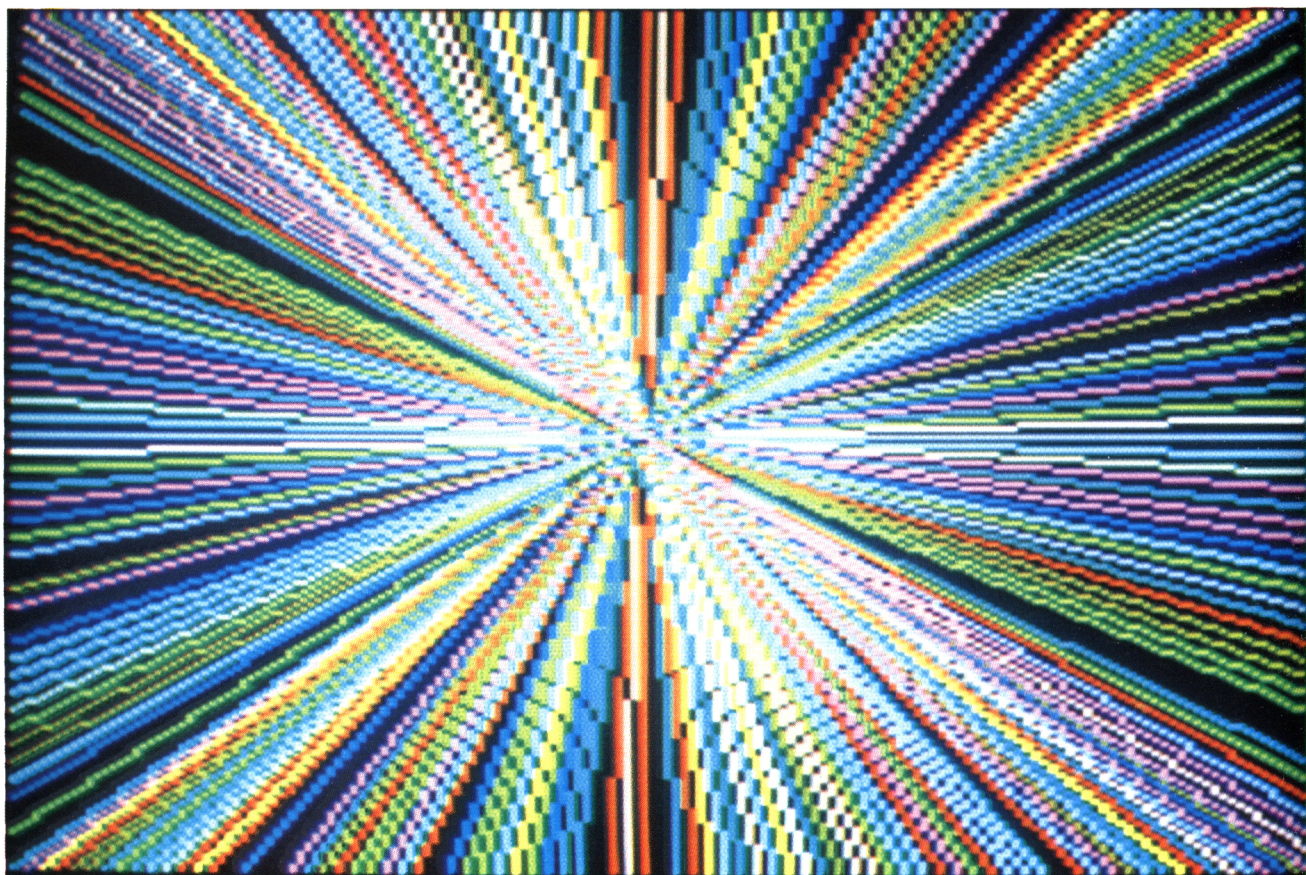


This is the program for the picture below. It has added sound effects. If you cannot hear them, turn up the volume control on your computer.

```
10 MODE 0
20 BORDER 0: PAPER 5:CLS
30 FOR X=0 TO 640 STEP 8
40 C=(RND*12)+1
50 PLOT X,10,C
60 DRAW 640-X,400
70 SOUND 1,X/C,2
80 NEXT X
90 FOR Y=0 TO 400 STEP 8
100 C=(RND*12)+1
110 SOUND 1,X/C,2
120 PLOT 0,Y,C
130 DRAW 640,400-Y
140 NEXT Y
150 IF INKEY$="" THEN 150
160 MODE 1:BORDER 1
170 PAPER 0:CLS:PEN 1
```



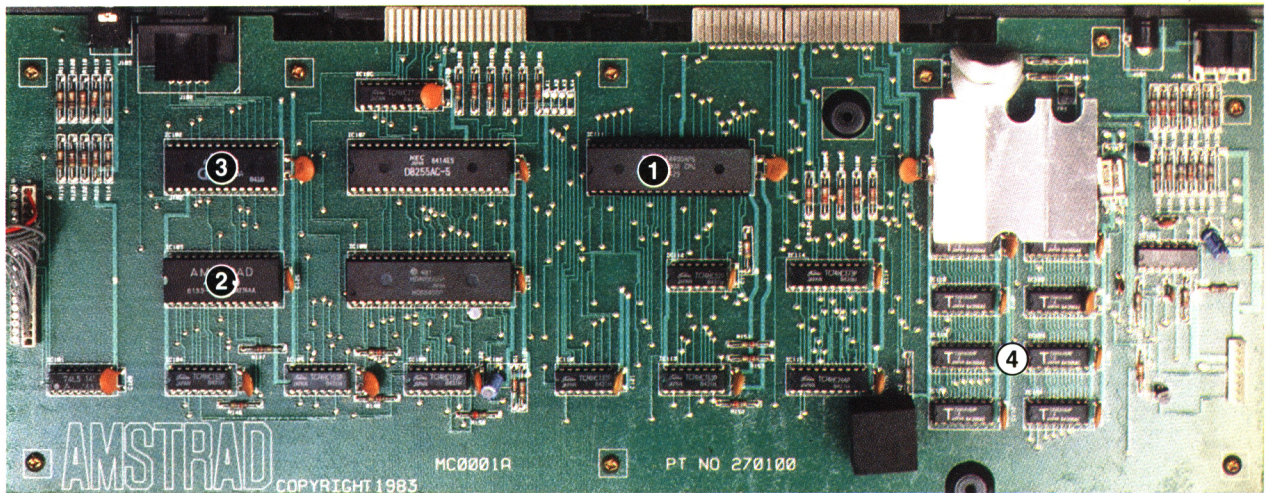
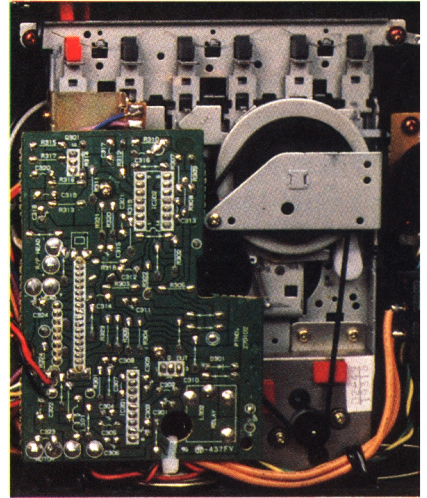
When you've run this program, you won't see the word READY on the screen. This is because of line 150. You can add a line like this to any program. To end the program, press any letter key.





# Inside Your Computer

Inside your computer are a number of chips mounted on a circuit board. While it is working, your computer is busy sending electronic signals to and from the chips. These signals travel along the metal tracks printed on the board. Each set of chips has a particular job to do.



## 1. The CPU chip

The most important chip is the CPU (Central Processing Unit). It is the 'general manager' of the computer and controls all the other chips. The CPU for the Amstrad CPC464 is the Z80 chip. Several other makes of home computer also use the Z80 chip as their CPU.

## 2. The ROM chip

When it is at work your computer does not use BASIC. It uses a special language known as MACHINE CODE. You can write programs in machine code but it is more complicated than using BASIC, especially for beginners.

The ROM chip programs the computer to understand the BASIC instructions and programs that you type in. ROM

Inside your computer:

1. The CPU chip, 2. The ROM chip, 3. The sound chip, 4. The RAM chips.

**\*MORFAX\***

Machine code runs faster than BASIC. Most arcade games are programmed in machine code to take advantage of this.



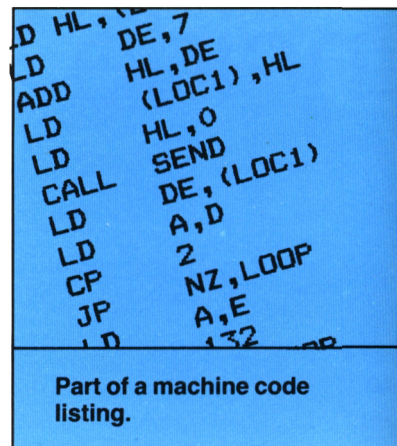
stands for READ ONLY MEMORY. The ROM translates the BASIC words and instructions you type in into the machine code that your computer understands.

### 3. The sound chip

This chip, the AY 8912, is responsible for producing the sound effects that you program into the computer.

### 4. The RAM chips

The RAM chips store the programs that you type in. RAM stands for Random Access Memory. The RAM chips store your programs only while the computer is switched on.



## The memory store

The programs that you feed into your Amstrad CPC464 are stored in the RAM chips. The longer and more complicated the program, the more space they take up. The amount of space available to store programs is measured in kilobytes or K.

Your computer has eight RAM chips. Each can store 8K of information, so your Amstrad CPC464 has 64K of memory. As with most computers, not all of this can be used to store your programs.

One kilobyte of memory can store about five hundred BASIC words and instructions. This may sound like a lot but programs can and often do fill nearly all the memory storage space in RAM. Because of this it is most important to use the memory space carefully. One reason why BASIC is more like a code than a language (such as English or French) is to save memory space.

Short programs like the ones in this book won't fill the memory space of your Amstrad CPC464. But saving memory space is an important part of BASIC programming, and as you will see, it can save typing time too.

#### \*MORFAX\*

There are all sorts of different computer languages. BASIC is often used for teaching beginners about programming. Its name stands for Beginners All Purpose Symbolic Instruction Code. Other computer languages include:

COBOL which is used mainly by businesses and offices, with larger computer systems known as MAINFRAME COMPUTERS.

PASCAL which is used mainly by small businesses and organisations to keep records and statistics.

#### \*MORFAX\*

A kilobyte (1K) is made up of one thousand and twenty-four BYTES. A byte is a binary number made of eight BITS (0 or 1), eg 11000101.



# Variables

## LET A stand for . . .

Certain numbers and words may be used more than once in your programs. But there is no need to keep on typing in the same information over and over again. Instead, you can store it, using the BASIC word LET.

To see how this works, type in:

```
10 LET A=9.872
```

What you have done here is to give the number 9.872 the name A. Your Amstrad CPC464 now stores this number in its memory as A. Every time you want to use 9.872 now, don't say 9.872, just say A. Add this next line:

```
20 PRINT A
```

and run the program to see how this works. You can use A as often as you like. Now add these lines:

```
30 PRINT A+A  
40 PRINT A*A  
50 PRINT A*10
```

and run the program. Typing A is easier than typing out the number 9.872. Although the number 9.872 is used six times in this program, it only has to be stored once in your Amstrad CPC464's memory. This helps to save memory space.

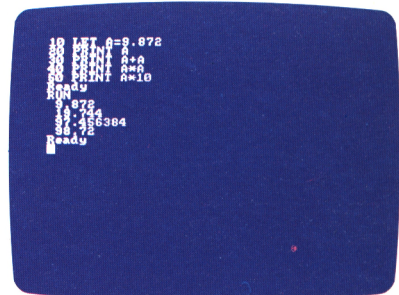
## More variable names

The letter A was used to stand for 9.872 in the last program. You can use any letter of the alphabet that you like, as long as you say what number it stands for at the beginning of your program.



**\*MORFAX\***

You can leave out the word LET if you want. Just type in A=10 instead of LET A=10.





```
10 LET Z=2.25
20 LET Y=7.575
30 PRINT Z
40 PRINT Y
50 PRINT Z+Y
Ready
RUN
2.25
7.575
9.825
Ready
```

You can also use more than one letter at a time. Type in:

```
10 LET Z=2.25
20 LET Y=7.575
30 PRINT Z
40 PRINT Y
50 PRINT Z+Y
```

and run it. Here again, why not add some lines of your own using Z and Y?

The letters that take the place of the numbers in your programs are called VARIABLES. Z and Y are examples of VARIABLE NAMES.

```
10 LET Z=5000
20 LET Y=256
30 PRINT Z
40 PRINT Y
50 PRINT Z+Y
Ready
RUN
5000
256
5256
Ready
```

The word 'variable' means something which can be changed. Look at that last program which you typed in. Change the first two lines like this:

```
10 LET Z=5000
20 LET Y=256
```

Now, when you run your program, your Amstrad CPC464 stores different numbers as Z and Y. You don't need to change the rest of the program to get a different result. Variables have many uses.

## Longer variable names

Letters like A and B or X and Y don't mean very much on their own. But variables can also have names longer than just single letters. This lets you choose names that describe what the variable stands for. When you read your listings, you should then be able to see what each variable stands for.

The next program shows you how to use longer variable names. It works out how many hours there are in two weeks. The number of days is multiplied by the number of hours in each day. Type in:

```
10 LET DAYS=14
20 LET HOURS=24
30 PRINT DAYS*HOURS
```

Run the program.

Longer variable names make your listings easier to understand.

```
10 LET DAYS=14
20 LET HOURS=24
30 PRINT DAYS*HOURS
Ready
RUN
336
Ready
```



Now try to change the program so that it works out how many hours there are in three weeks. If you're feeling very brave, try working out how many minutes there are in one year! Any answers?

**\*MORFAX\***

Amstrad BASIC allows your variable names to be up to forty letters long. They cannot include spaces. This line:

```
10 LET EXAM RESULT=0
```

would give a 'Syntax error' message.

Many programmers prefer to type variable names in small letters, and you may see listings which print them like this. If you find it easier to use capital letters instead, you can.

Variable names  
can be either  
UPPER CASE  
or lower case.



## Microshopping

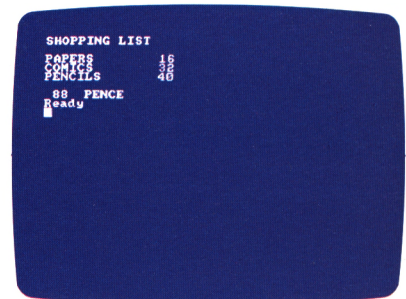
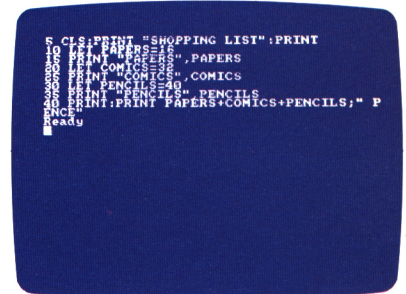
The program below adds up a list of groceries. The variables store the price of each item. Type it in and run it:

```
10 LET PAPERS=16
20 LET COMICS=32
30 LET PENCILS=40
40 PRINT PAPERS+COMICS+PENCILS
```

The screen display for this program is not very 'friendly'. It just prints a number. Add the following lines and see the improvement:

```
5 CLS:PRINT "SHOPPING LIST":PRINT
15 PRINT "PAPERS ",PAPERS
25 PRINT "COMICS ",COMICS
35 PRINT "PENCILS ",PENCILS
40 PRINT:PRINT PAPERS+COMICS+PENCILS;
" PENCE"
```

The number of hours in the day stays the same, but the price of goods changes. Changing numbers stored as variables without changing the rest of the program helps to cope with this.





# Storing words

Variables can also be used to store words, sentences, or just parts of words, using LET. Type in:

```
10 LET W$="AMSTRAD CPC464"  
20 PRINT W$
```

Run the program. That dollar sign (\$) after the variable is very important. It tells your computer that you want to store words or strings. Words or strings to be stored as variables must be stored using quotation marks, or your Amstrad CPC464 will tell you that you've made a syntax error.

Now that it is stored in the memory, every time you want to use the words AMSTRAD CPC464, don't say AMSTRAD CPC464, say W\$. Add these lines to your program to see this working:

```
20 PRINT "I AM PROGRAMMING MY ";W$  
30 PRINT "THE ";W$;"HAS A 64K MEMORY"  
40 PRINT "I AM LEARNING TO PROGRAM  
MY ";W$
```

Longer variable names used with words also help to describe the kind of information stored. For example, type in:

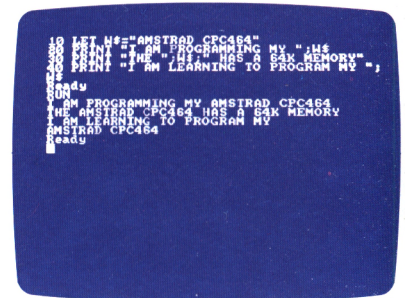
```
10 LET NAME$="MIKE ROE"  
20 LET ADDRESS$="64 K MEMORY LANE"  
30 LET TOWN$="CHIPPENHAM"  
40 PRINT "MY NAME IS ";NAME$  
50 PRINT "I LIVE AT ";ADDRESS$  
60 PRINT "MY HOME TOWN IS ";TOWN$
```

and run the program.



## \*MORFAX\*

Your computer will not accept BASIC words as variables. You cannot for instance say LET RUN=1, or LET RUN\$="HI".





# Editing Programs

When you want to change a line you can type in a completely new version, or you can EDIT the line already there. Type in:

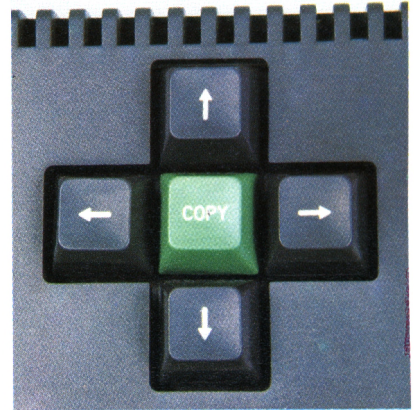
```
10 PRINT "I AM 12 YEARS OLD"  
20 PRINT "MY NAME IS ARNOLD"
```

Now press SHIFT and the key marked ↑. The cursor moves up one line. Do this again so that the cursor is over the 1 at line 10.

In the middle of the cursor keys (the ones with arrows on) is a green key marked COPY. Press this a few times and you will see that line 10 is copied out below the program. Do this until the cursor is over the 2. Press the COPY key, delete the 2 using the green DEL key, and type in 3 instead. Continue pressing COPY until the rest of the line is copied out. Press the ENTER key and list the program. You will find that the program now contains the 'edited' version of line 10.

Now, using SHIFT and the cursor keys, move the cursor to line 20. The key marked ↓ moves the cursor down; the key marked ← moves it left, and the key marked → moves it right. Copy out the line until it is in between the words IS and ARNOLD. Press the space bar and type in the word NOT. Copy out the rest of the line, press ENTER and list the program to see the new version of line 20.

If you have to type in a line with the same instructions as one that you've already typed in, type in the new line number and then move the cursor up to that line. Now COPY the rest of the line, and press ENTER.





## Using EDIT

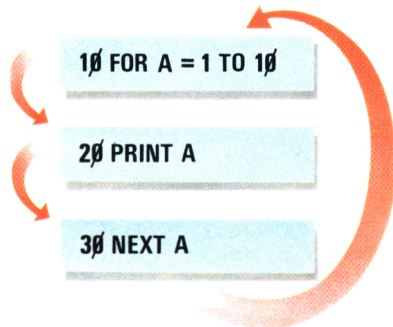
You can also change a particular line using the instruction EDIT. For example, to change line 20, you should type in the instruction:

EDIT 20

and press ENTER. Line 20 appears on the screen. Don't press the SHIFT keys, just use the cursor keys to move the cursor along the line and add any changes just as you would with a normal program line.

Loops are hard to understand at first, but don't worry. You'll get used to them.

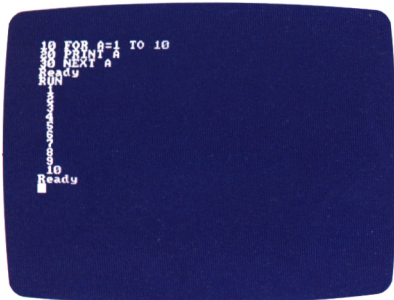
# Loops



So far you have only used variables to stand for one number at a time. However, you can also make a variable stand for several different numbers in one program.

The program below shows you how a variable can stand for the numbers 1 to 10. Type in the program and run it:

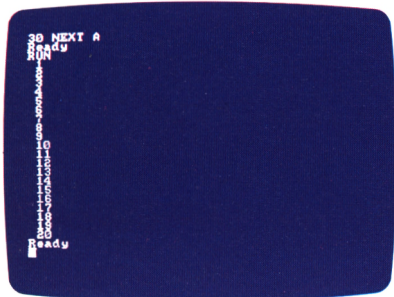
```
10 FOR A=1 TO 10
20 PRINT A
30 NEXT A
```



The program prints the numbers 1 to 10. In this program then, the variable A stands FOR the numbers 1 TO 10. Change line 10 to this:

```
10 FOR A=1 TO 20
```

Run the program. Now A stands FOR the numbers 1 TO 20. This is called a LOOP. Your Amstrad CPC464 begins by printing A as 1. This is the first value of A shown at line 10. It prints this value, then moves to line 30 which says NEXT A. This tells it to take the NEXT value of A. It returns to line 10 to see what this is. In this program it is 2, the 'next' number after 1 between 1 and 20. The program keeps on going round and round until A is equal to the last number mentioned at line 10. Once A reaches 20, the program stops.





# FOR . . . NEXT

Because this kind of loop uses the BASIC words FOR and NEXT, it is known as a FOR . . . NEXT loop. You cannot use the word FOR unless it is followed by the word NEXT later on in your program. FOR . . . NEXT loops have many uses.

## STEP

You can use a FOR . . . NEXT loop to count up in steps, using the word STEP. Type in:

```
10 FOR A=5 TO 60 STEP 5
20 PRINT A
30 NEXT A
```

Run the program. A now goes from 5 to 60 in steps of 5.



Another way of using STEP is to go up in steps of less than one, that is, in fractions. Type in:

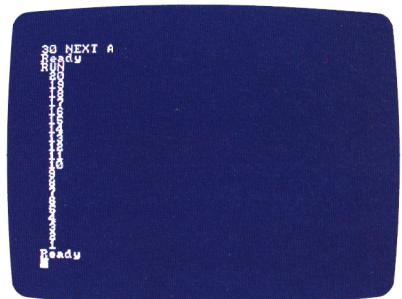
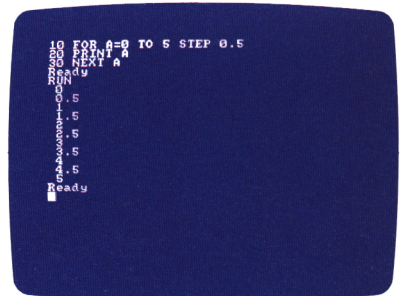
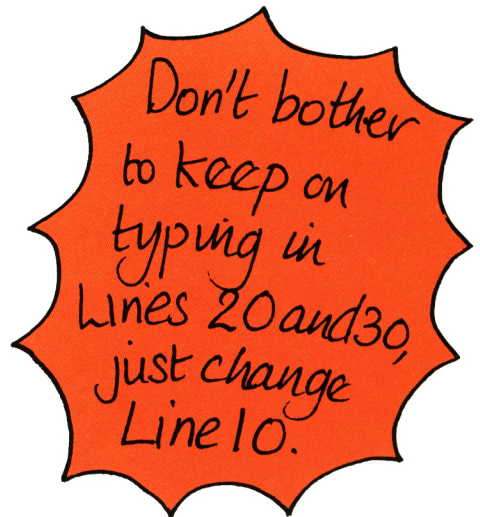
```
10 FOR A=0 TO 5 STEP .5
20 PRINT A
30 NEXT A
```

Run the program and you will see that your program now prints A not just as whole numbers, but as decimals too.

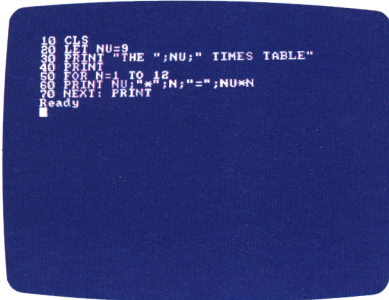
STEP can help you to count down as well as up. To do this you use a minus sign after STEP and start your loop with the highest number first. Try this to see how it works:

```
10 FOR A=20 TO 1 STEP -1
20 PRINT A
30 NEXT A
```

Have a go at changing the loop yourself now, using STEP with different numbers.



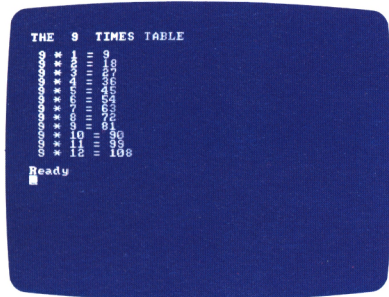




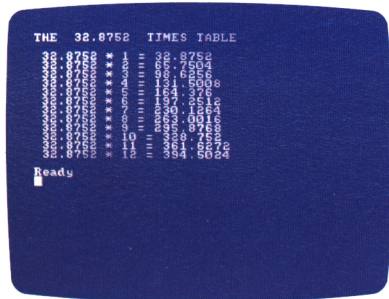
# Doing your tables

FOR . . . NEXT loops can print out tables very quickly. This next program uses a loop to print out the 9 times table.

Instead of just printing out the numbers on their own, which looks rather dull, the program uses PRINT with words and symbols. This makes the screen display clearer and more informative. Type in:

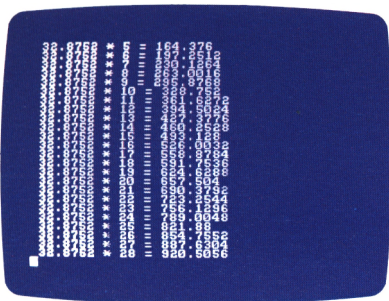


```
10 CLS
20 LET NU=9
30 PRINT "THE ";NU;" TIMES TABLE "
40 PRINT
50 FOR N=1 TO 12
60 PRINT NU;"*";N;"=";NU*N
70 NEXT N: PRINT
```



Run the program. This is quite good fun but does not really test your Amstrad CPC464 to the full. The 9 times table is really rather easy. How well do you know the 32.8752 times table? Why not get your Amstrad CPC464 to do it for you? Just change the 9 at line 20 and put 32.8752 instead. You'll be surprised at how quickly you get the answers!

Dont get rid of this program yet. Now we're going to make your Amstrad CPC464 work even harder!



## Scrolling and pausing

The programs that you've written so far have only taken up a small amount of the screen display space. Change line 50 to this:

```
50 FOR N=1 TO 60
```

Run the program and see just how quickly your Amstrad CPC464 can work. Unfortunately it moves almost too quickly. Once the screen has filled up with information, it starts to SCROLL, ie the display rolls up and off the top of your screen.

To stop this happening, press the red ESC key. This stops the program running and lets you see what is on the screen at the time. To continue with the program, press one of the letter or number keys.

### \*MORFAX\*

To break out of a program altogether, press ESC twice. This brings the Ready message back on to the screen.

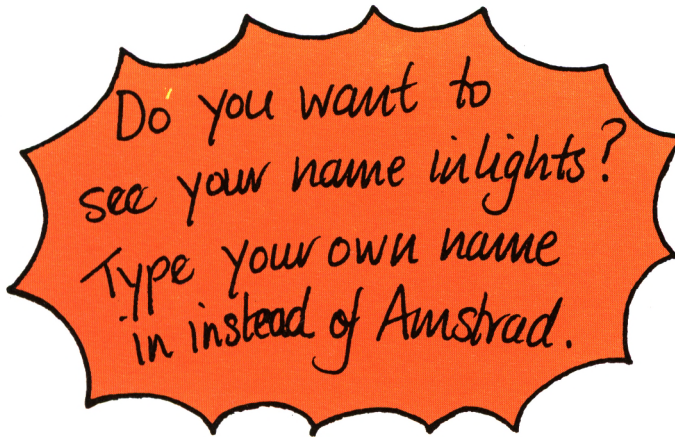


# Words and patterns

Loops can be used with words too. They can tell your Amstrad CPC464 to print a word a certain number of times. Try this:

```
10 LET W$="AMSTRAD"  
20 FOR L=1 TO 80  
30 PRINT W$;  
40 NEXT L
```

Why not add some colour to this program?



**\*MORFAX\***

A FOR . . . NEXT loop which does nothing can be used to slow down the screen display. Put one in at line 35 like this:

```
35 FOR X=1 TO  
500: NEXT X
```

and run the program again to see how it is affected.

**\*MORFAX\***

Did you know that you can program the number keys on the numeric keypad? Type this in now:

```
KEY 128, "PRINT"
```

and press ENTER. Now press key 0 (on the keypad). As you can see, the word PRINT appears on the screen. This can make it easier to type in your programs. Or try typing this in:

```
KEY 129, "CLS:LIST"+CHR$(13)
```

Now, whenever you press key 1, the screen clears and your program will list. The instruction +CHR\$(13) has the same effect as pressing the ENTER key.

The other number keys can be programmed in the same way. For key numbers 2 to 9, use numbers 130 to 137 after the word KEY.





# Chance

So far you've chosen all the numbers in your programs. Using the BASIC word RND, you can let your Amstrad CPC464 pick the numbers. RND stands for random. Random means 'by chance'. Let's see how this works. Type in:

```
10 PRINT RND*10
```

and run the program a few times.

Your Amstrad CPC464 will pick at random a number between 0 and 10, including long decimal fractions.

Now change line 10 to this:

```
10 PRINT INT(10*RND)
```

Run the program a few times. Can you see what the word INT does? It rounds the numbers down so that only whole numbers get printed. This way you will get numbers from 0 to 9, but never ten. If you want whole numbers from 1 to 10, change line 10 like this:

```
10 PRINT INT(10*RND)+1
```

To print out a column of random whole numbers, add these lines to your program:

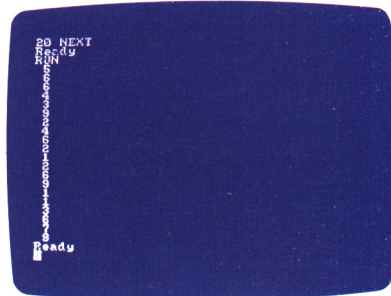
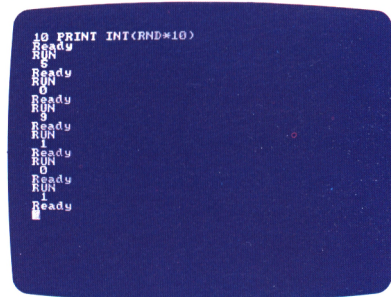
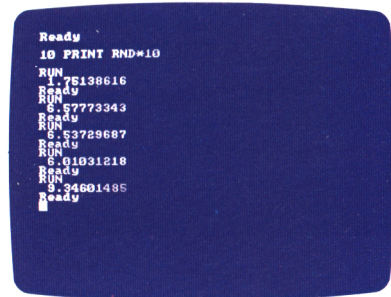
```
5 FOR R=1 TO 20  
20 NEXT
```

## Random colours

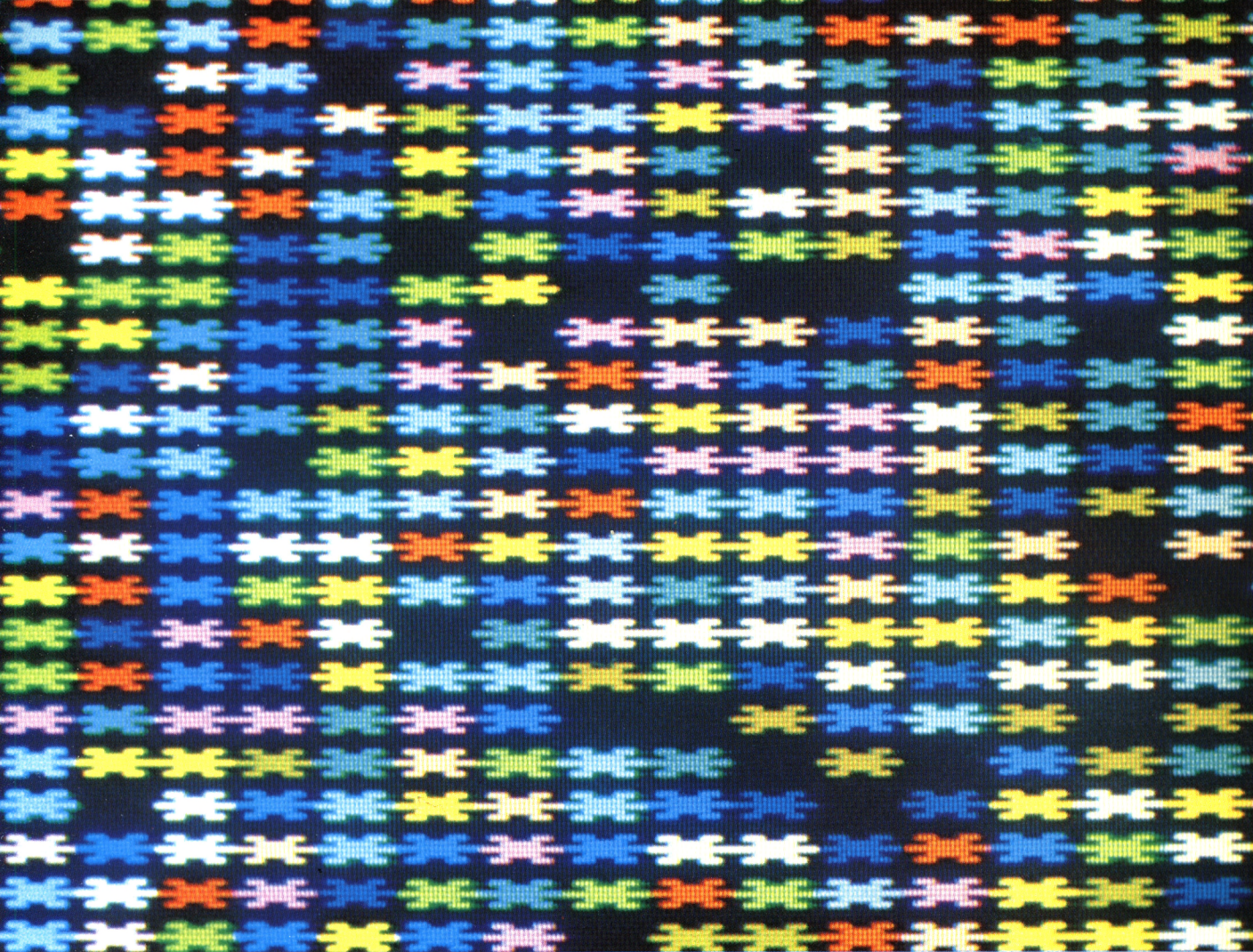
You can use RND to display colours at random. The next program shows you how to do this.

Lines 20 and 30 set the screen display colours. Line 50 then chooses, at random, the colour of the shape.

To reset the screen when the program is over, press any key.







Type in:

```

10 CLS
20 MODE 0
30 BORDER 0:PAPER 5:CLS
40 FOR X=1 TO 1000
50 C=INT(15*RND)
60 PEN C
70 PRINT "*";
80 NEXT
90 IF INKEY$="" THEN 90
100 MODE 1: BORDER 1:PAPER 0
110 PEN 9:CLS

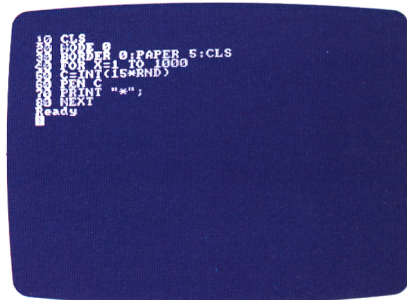
```

You can obtain the full range of colours available in MODE 0 by using the word INK. Using INK with RND produces interesting results. Add these lines:

```

45 INK RND*15,RND*26
100 MODE 1
110 BORDER 1:INK 0,1: INK 1,24
120 PAPER 0:PEN 9:CLS

```

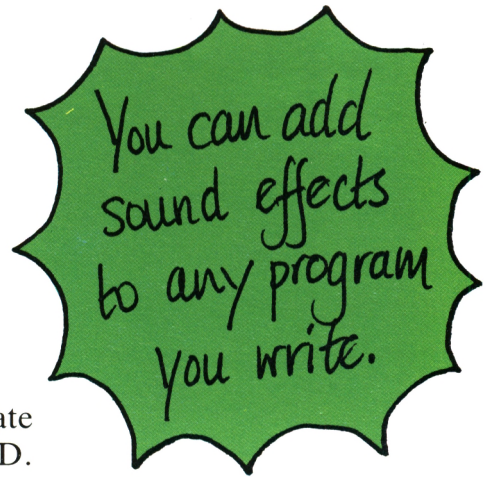


#### \*MORFAX\*

In MODE 0 you can display up to sixteen colour effects at one time. The Amstrad CPC464 offers twenty-six colours in all, but you cannot see all of these on the screen at once.



# Sound Effects



Your Amstrad CPC464 can be used to play music and create sound effects. To do this you use the BASIC word SOUND. Just as each colour has its own code number, so do notes and sounds. Type in:

```
SOUND 1,478,200,15
```

and press ENTER. This plays the note Middle C for two seconds.

The first number following the word SOUND stands for the CHANNEL in use. You can use these to produce several sounds at once, but for the time being stick to Channel 1.

The second number stands for the note you are playing. The manual that comes with your computer lists what numbers you need to use for each particular note.

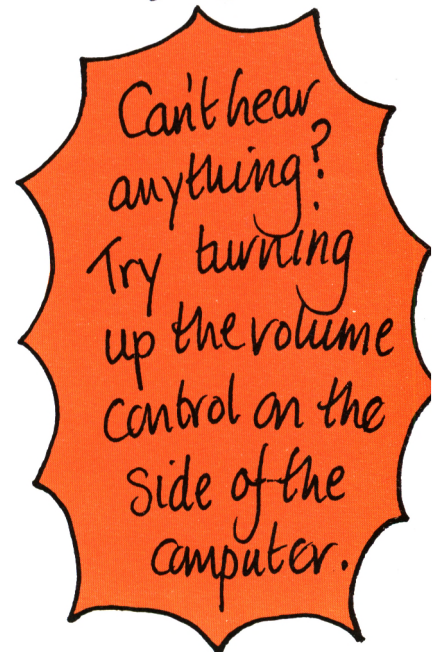
The third number stands for the length of time that the note will be played. This is measured in one-hundredths of a second, so 200 means two seconds.

The fourth number stands for the volume of your note. Fifteen is the loudest and nought is the quietest (it produces silence).

Let's try another note now. Type in:

```
SOUND 1,239,500,15
```

This plays the note C again but this time it is one octave higher. Try using different numbers. Don't worry if you do not understand exactly what they stand for, just have fun!





### Loopy sounds

Using loops with SOUND can create some interesting effects. Try this program:

```
10 FOR N=0 TO 4095
20 SOUND 1,N,1,15
30 NEXT N
```

This plays every single tone available for one hundredth of a second each.

When you've run this a few times, change line 10 to this:

```
10 FOR N=0 TO 4095 STEP 5
```

### Up and down

You can make your sounds go down by using a minus loop. Try this:

```
10 FOR N=4095 TO 0 STEP-20
20 SOUND 1,N,1,15
30 NEXT N
```

### Looping the loop

Putting one loop inside another will play a particular sound effect several times over. This next program could be one of those laser blasters!

```
10 FOR G=1 TO 5
20 FOR N=300 TO 20 STEP-5
30 SOUND 1,N,.9,15
40 NEXT N
50 NEXT G
```

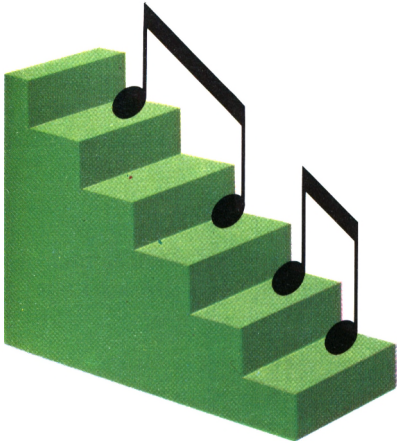
### Random sounds

Using RND with SOUND is also fun. Try this:

```
10 FOR G=1 TO 500
20 N=INT(600*RND)+200
30 SOUND 1,N,10,15
40 NEXT G
```

To hear a noise like a swarm of bees, just change the 200 here to 1000!

The lower the second number after SOUND, the higher the note played.



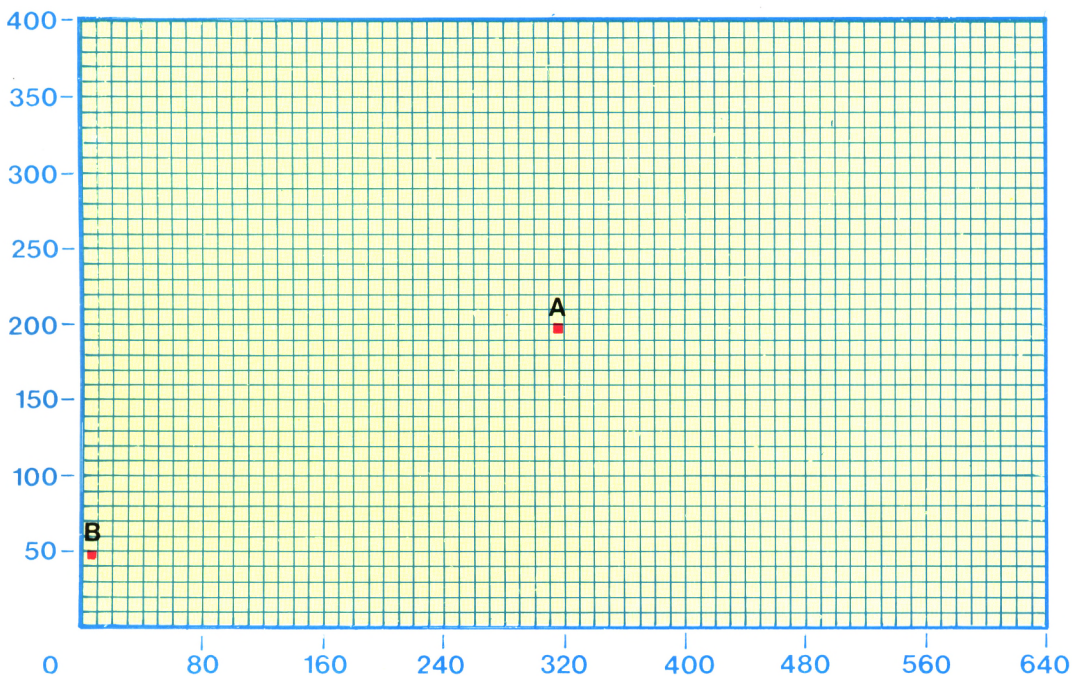
A loop in a loop is called a nested loop.



# Drawing

You can draw shapes and patterns with your Amstrad CPC464. Start with some simple lines and shapes and experiment with them, and you will soon learn how to produce striking results.

When you draw with your Amstrad CPC464, the screen is divided up like a grid or sheet of graph paper. The grid measures 640 by 400 blocks. You use numbers to say where on the screen you want your lines and shapes to go.



Each block or PIXEL on the grid is identified by two numbers called co-ordinates. To find the numbers or co-ordinates for each square, you count the blocks across from the left first and give that number. You then count the blocks up from the bottom, and give that number next.

In the diagram above, Point A is 320 blocks along and 200 blocks up. Point B is 10 blocks along and 50 blocks up.

The co-ordinates of A are 320,200  
The co-ordinates of B are 10,50

## \*MORFAX\*

Each point or block on the screen is known as a PIXEL.

In computing and maths the squares going across are said to be on the X axis. Those going up are on the Y axis.



# Using PLOT

To draw a point on the screen, use the word PLOT followed by three numbers. The first two numbers are the co-ordinates of the point. Remember, give the X co-ordinate first (saying how many blocks across the grid your point is). The next number, the Y co-ordinate, says how far up the screen the point is. The third number is used to set the colour of the point. Once you have set this you do not need to use the third number again until you wish to change the colour being used. To PLOT the two points you have just looked at in MODE 0, type in:

```
10 MODE 0
20 PLOT 320,200,1
30 PLOT 10,50
```

Run the program. Now change line 10 to make it run in MODE 1, and then in MODE 2. As you will see, the pixels are smaller in these modes. Try plotting some points for yourself in all three modes, just to get used to plotting points.

## \*MORFAX\*

Listings in MODE 0 can be hard to read. To reset the screen to the standard MODE 1 display, program keypad key 0 like this. Type in:

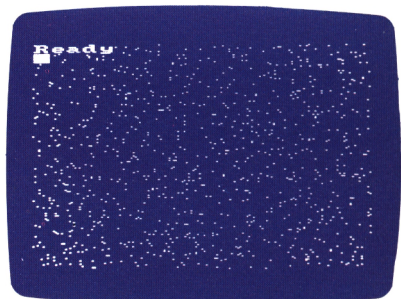
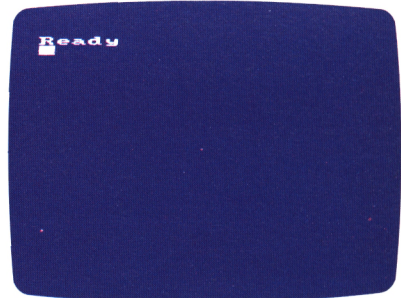
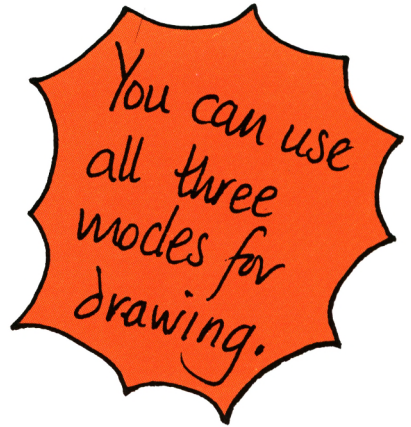
```
KEY 128,"INK 0,1:INK 1,24:MODE 1:
LIST"+CHR$(13)
```

Now press ENTER. Before you list your program now, just press keypad key 0.

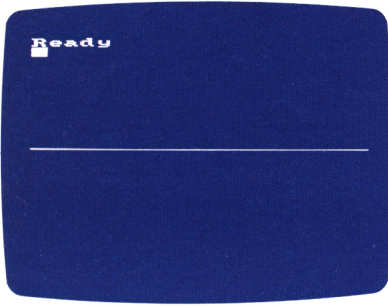
## Random points

You can use RND to plot points at random. If you use a loop as well, you can fill the screen with random points. Type in:

```
10 MODE 0
20 FOR N=1 TO 1000
30 LET X=INT(RND(1)*640)
40 LET Y=INT(RND(1)*400)
50 PLOT X,Y,1
60 NEXT N
```







## Drawing lines

To draw lines you use the BASIC word DRAW. First set a starting point with PLOT. Then use DRAW followed by the co-ordinates of the end point of the line. For example, this program draws a straight line across the screen:

```
10 MODE 0
20 PLOT 0,200,1
30 DRAW 640,200
```

Do you think you could draw a straight line that goes up the screen?

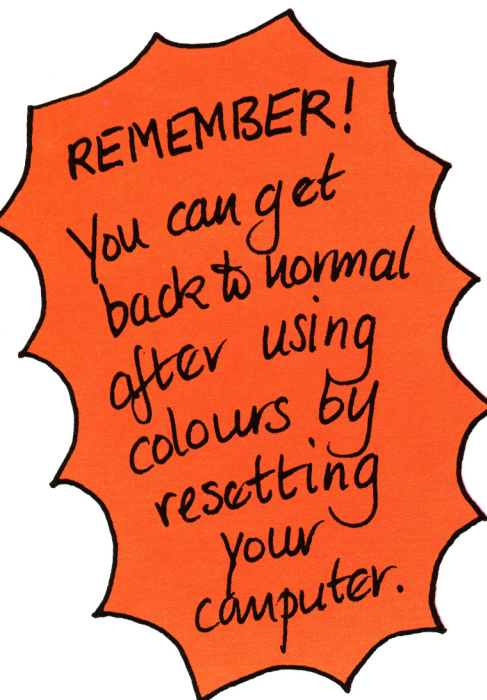


## Adding colour

To add colour to your programs you have to use a third number after your PLOT or DRAW statements. The chart below shows you what numbers to use in MODE 0.

### \* MODE 0 Colour Chart \*

0 Blue	8 Cyan
1 Bright Yellow	9 Yellow
2 Bright Cyan	10 Pastel Blue
3 Bright Red	11 Pink
4 Bright White	12 Bright Green
5 Black	13 Pastel Green
6 Bright Blue	14 Yellow/Blue
7 Bright Magenta	15 Pink/Sky Blue



This program draws two diagonal lines in different colours. Type in:

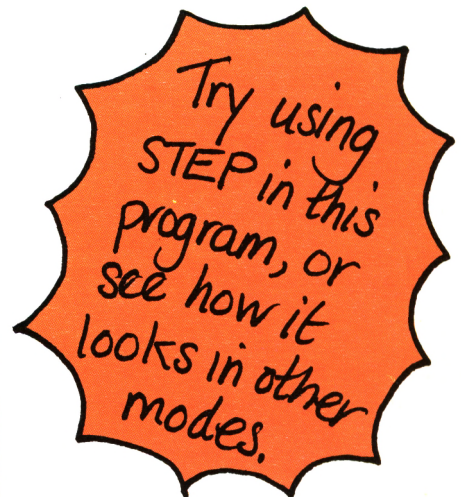
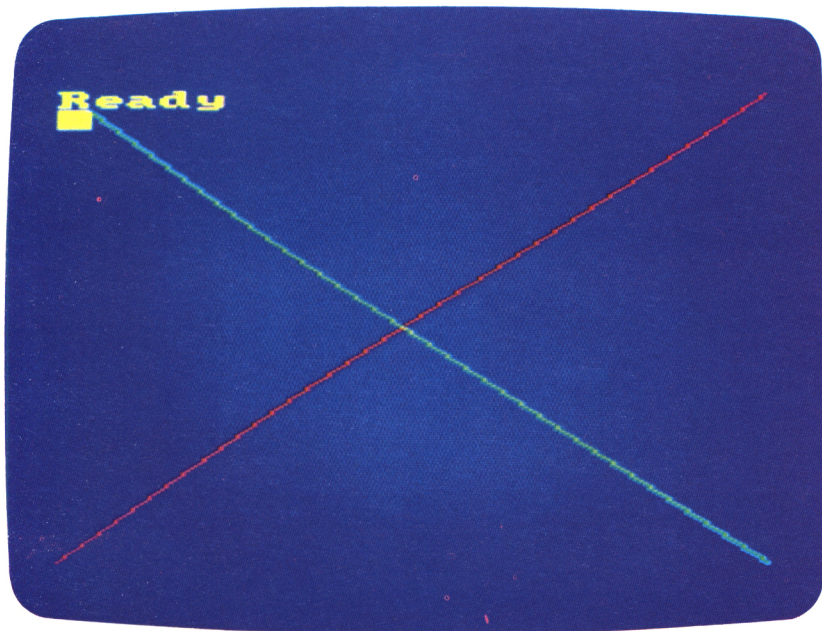
```
10 MODE 0
20 PLOT 0,0,3
30 DRAW 640,399
40 PLOT 640,0,12
50 DRAW 0,400
```



# Loopy lines

This next program uses a loop to draw a pattern of lines which fill the screen. Colour is added at random. Type in:

```
10 MODE 0
20 FOR Y=0 TO 400
30 C=INT(RND*16)
40 PLOT 0,Y,C
50 DRAW 640,Y
60 NEXT Y
70 IF INKEY$="" THEN 70
```



# Drawing a box

To draw a rectangle, you use DRAW like this:

```
10 MODE 0
20 PLOT 50,50,1: DRAW 590,50
30 DRAW 590,350: DRAW 50,350
40 DRAW 50,50
```

Notice that you only need to say what colour your shape is going to be once, at line 20. Any other lines or points plotted after this will be in the same colour.

Can you draw a smaller rectangle to fit inside this one? Perhaps you could even try a triangle.

## \*MORFAX\*

Lines 20 and 30 of this program are known as MULTI-STATEMENT lines. This is because these lines contain two sets of instructions which could go on separate lines. Instead they are joined together with a colon, the : sign.



# More boxes

This next program uses a loop to draw a pattern based on rectangles. It also adds colour and sound to make the results more impressive.

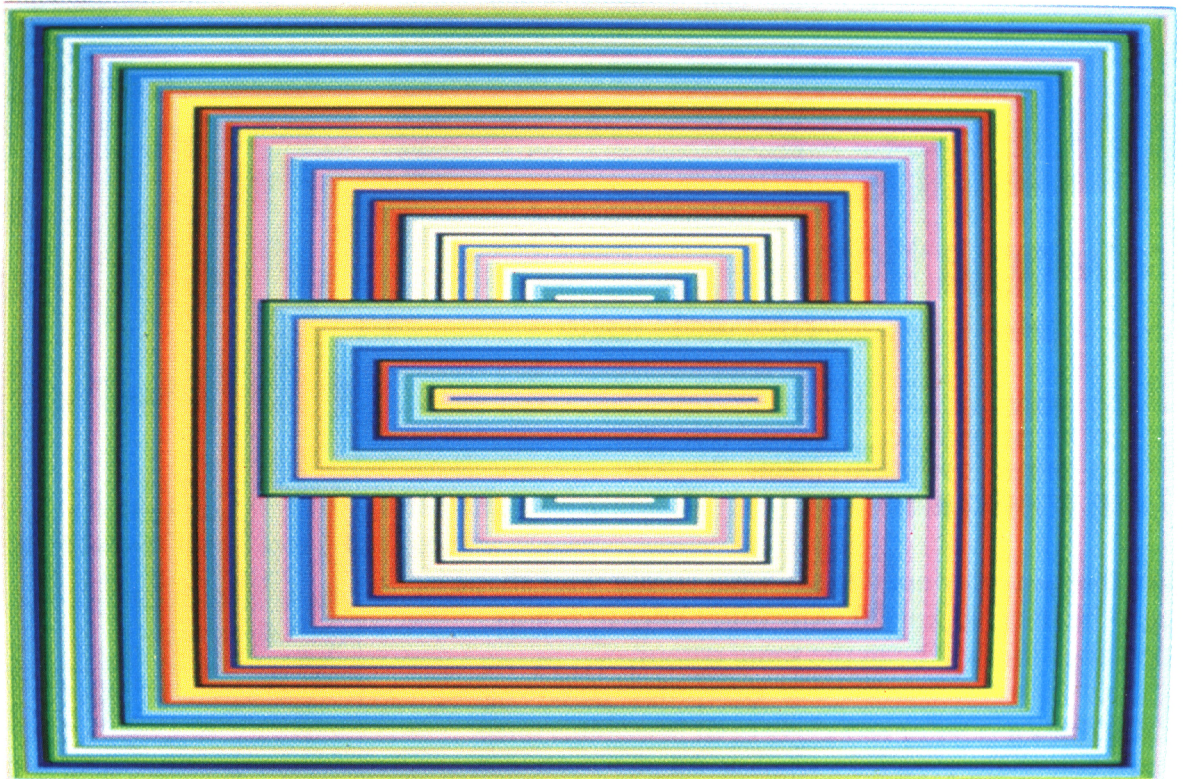
```
10 MODE 0
20 BORDER 0
30 Y=0:E=640:T=400
40 FOR X=0 TO 640 STEP 4
50 C=INT(RND*16)
60 PLOT X,Y,C:DRAW X+E,Y
70 DRAW X+E,T:DRAW X,T:DRAW X,Y
80 Y=Y+2:E=E-8:T=T-2
90 SOUND 1,X,1,15
100 NEXT
110 IF INKEY$="" THEN 110
120 INK 0,1:INK 1,24
130 BORDER 1:MODE 1
```

## \*MORFAX\*

Drawing pictures using a computer is known as COMPUTER GRAPHICS.

To draw very detailed pictures, the squares or pixels which make up the display grid have to be very small. When this happens, the computer is said to have high resolution graphics. With low resolution graphics you cannot produce such fine detail.

MODE 2 is the high resolution graphics mode on the Amstrad CPC464.



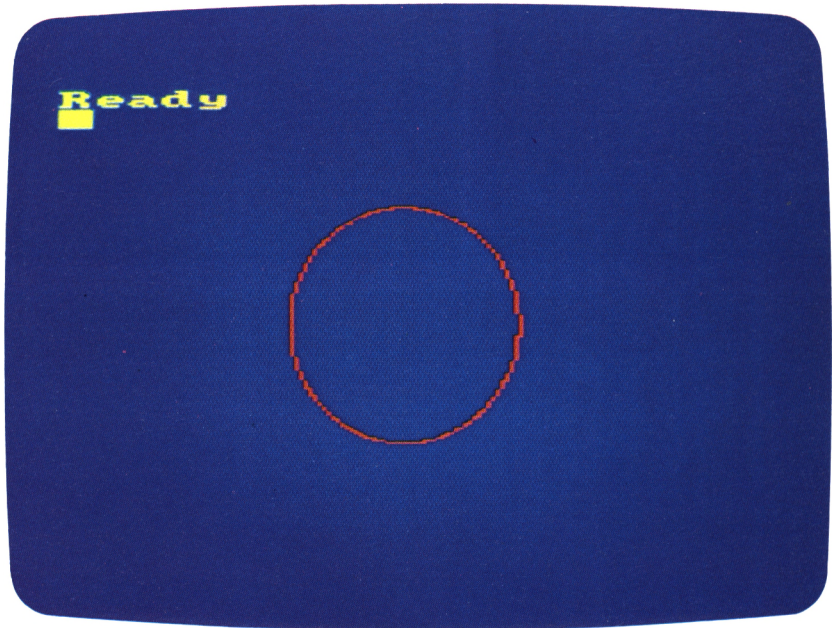


# Circles

Drawing circles with your Amstrad CPC464 is quite a complicated process. Don't worry if you don't understand the mathematics involved in the program below. Just type it in and run it.

```
10 MODE 0
20 FOR A=1 TO 360
30 DEG
40 R=100:X=320:Y=200
50 PLOT X+R*COS(A),Y+R*SIN(A),1
60 NEXT A
```

Line 40 is a multi-statement line. Combining several instructions saves memory space but can make your listings hard to read.



To define a circle, you say where the centre point is, and how far away from it the line is to be drawn. The distance from the circle's centre to its edge is called the radius.

In this program, line 40 sets the radius of the circle, R, to 100. X and Y are the co-ordinates of the circle's centre point. Try changing these numbers to draw circles elsewhere on the screen.

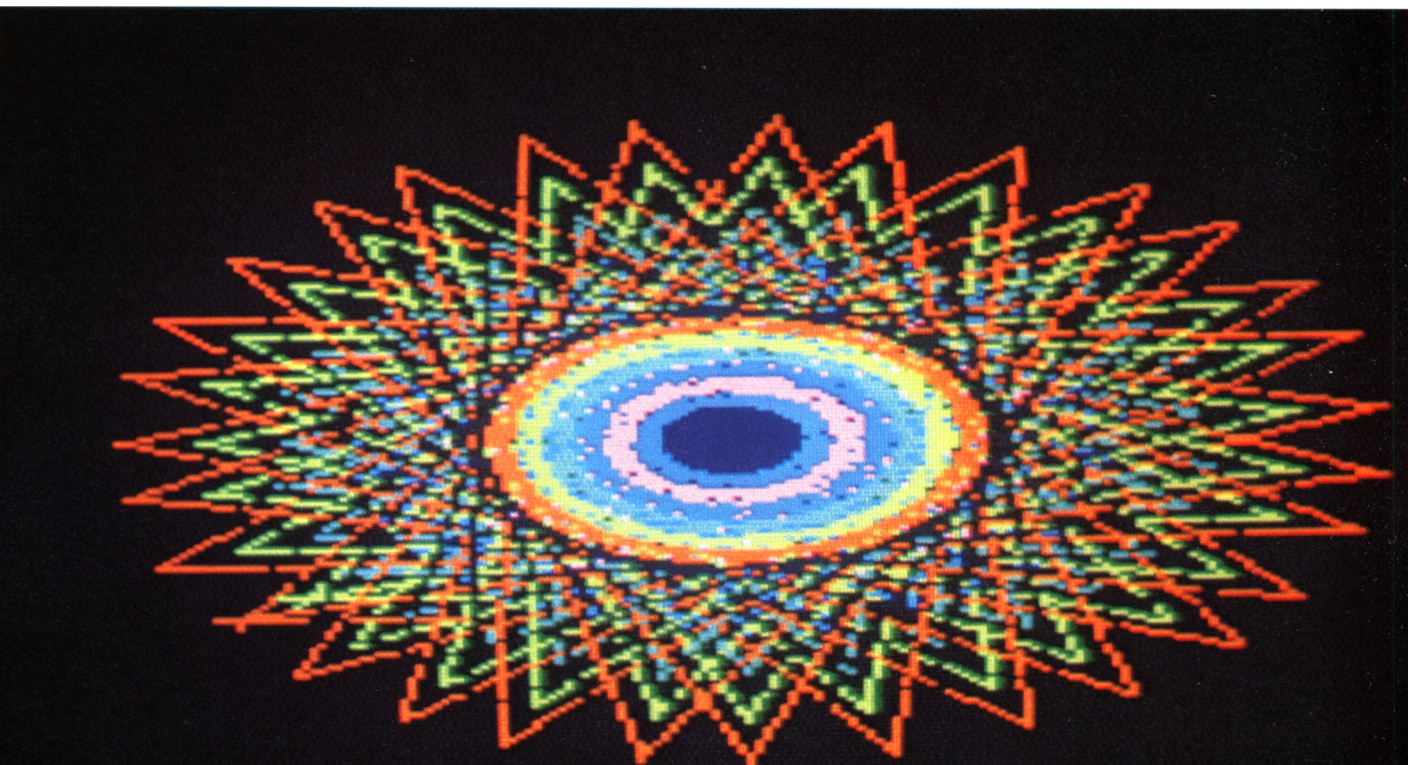
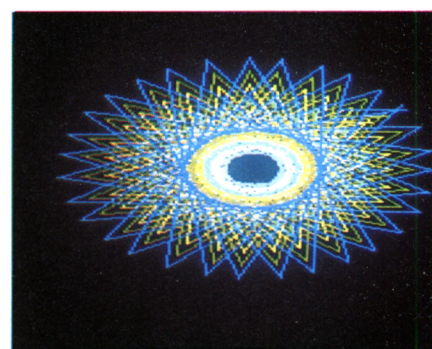
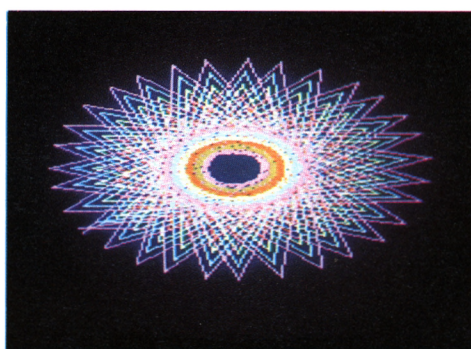
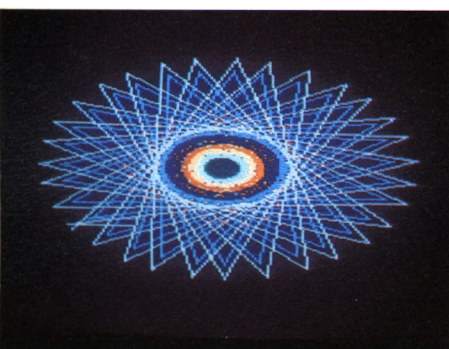




Now try this program. It uses loops, PLOT and DRAW with the mathematical formulas used to draw a circle, and makes an interesting pattern. Type in:

```
10 MODE 0: BORDER 0
20 INK 0,0: PAPER 0: CLS
30 FOR R=50 TO 180 STEP 20
40 A=10: C=INT(RND*16)
50 PLOT 320+R*COS(A),200+R*SIN(A)
60 FOR J=1 TO 31
70 DRAW 320+(R*1.75)*COS(A),200+R*SIN(A),C
80 A=A+15:NEXT J
90 NEXT R
100 IF INKEY$="" THEN 100
110 MODE 0: INK 0,1: INK 1,24
120 BORDER 1: PAPER 0: CLS: PEN 1
```

Change the number after STEP in line 30 and see what difference it makes.





# Loading and Saving

When you save a program on cassette, you are making your own software.

By now you may have written some programs which you would like to keep so that you can use them again. You don't need to type them in every time. Instead you can record or SAVE them on cassette tape.

When you want to use the program again you feed or LOAD it into your Amstrad CPC464. To do this you use the computer's own built-in cassette unit. You may already have ready-made programs, known as SOFTWARE, on cassette tapes. You will need a blank cassette tape now for saving and loading your own programs.

Use the tape counter to keep a record of where your programs are stored on your tapes.





# Saving programs

Type in a short program now, any of the ones in this book will do. Put a tape in your cassette unit. Make sure that it is wound past the leader tape at the beginning of the cassette. You now have to tell your Amstrad CPC464 that you want to save the program. Type in on a separate line:

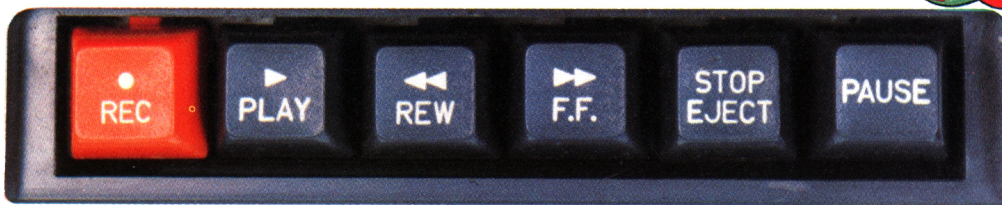
```
SAVE "PROGRAM1"
```

We have called the program PROGRAM1 in this case, but you could give it a name of your own which describes the program.

Press ENTER. You will see this message on the screen:

```
Press REC and PLAY then any key
```

Press the RECORD and PLAY buttons on your cassette unit and then press any of the letter keys.



You will now see this message on your screen:

```
Saving PROGRAM1 block 1
```

When the program has been saved, ie recorded successfully, you will see the 'Ready' message. The tape will have stopped automatically, but it is still a good idea now to press the STOP button on your cassette unit.

## \*MORFAX\*

When you save a program on tape, your computer translates all the words in your program into a series of electronic pulses. These produce a strange noise. You can hear them if you try playing one of your tapes on a normal tape recorder. Because programs can be turned into sounds, computer programs can be transmitted in the same way as radio programmes or phone calls.

*Give your programs sensible names to help you identify them later.*

*Don't give the same name to two programs. It could be confusing.*



# Loading Programs

When you want to use a program that you've saved on tape, you have to **LOAD** it in. Put the tape in the cassette unit. Remember to rewind it. Now type in:

```
LOAD "PROGRAM1"
```

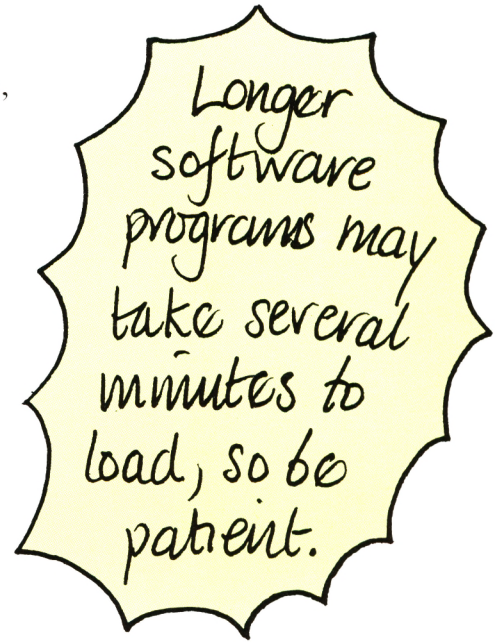
Remember to put the name of your program inside quotation marks. You will see the message:

```
Press PLAY then any key
```

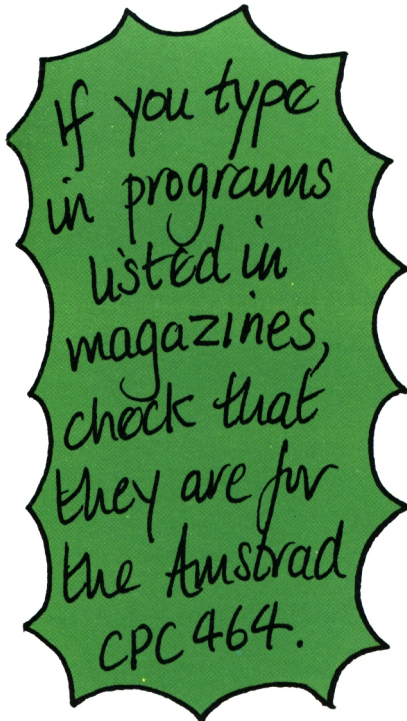
Press the **PLAY** button on your cassette unit and then one of the letter keys. After a while you will see this message:

```
Loading PROGRAM1 block1
```

When the program has loaded you will see the 'Ready' message. Press the **STOP** button on your cassette unit. You can now run your program in the usual way.



## Taping tips



### Loading software

If you're loading in games or other software, always switch the computer off and on again before you do so.

Instructions from previous programs can interfere with your games.

### Not sure of the program's name?

If you can't remember what programs you have on your tape, type in **LOAD ""** and press **ENTER**. This loads in the first program the computer finds on your tape.

### Can't find a program?

If you want to load a program which is in the middle of a tape and you have rewound your tape back to the beginning, your computer looks at the name of each program in turn. If the name it finds is not the program you want, don't worry. Don't press any keys or buttons. After a short pause it will carry on automatically, looking for the next program on the tape until it finds the one you want.



## Can't remember what's on a tape?

To list all of the programs on a particular tape, just type in LOAD " followed by a program name you know is not on the tape. As your computer reads through the tape you will see the name of each program printed on the screen.

## Using RUN

If you type in the instruction RUN followed by the program name in quotation marks and then press ENTER, your program runs automatically.



# Programs to save

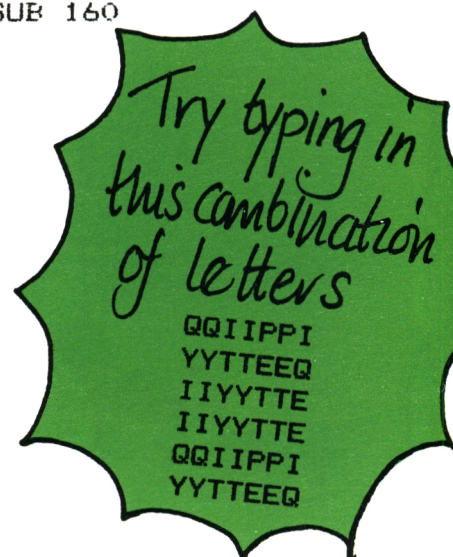
This program turns your Amstrad CPC464 into a musical instrument. Type it in:

```
10 LET A$=INKEY$
20 IF A$="S" THEN END
30 IF A$="Q" THEN T=239:PRINT"C#";:GOSUB 160
40 IF A$="W" THEN T=225:PRINT"C#";:GOSUB 160
50 IF A$="E" THEN T=213:PRINT"D#";:GOSUB 160
60 IF A$="R" THEN T=201:PRINT"D#";:GOSUB 160
70 IF A$="T" THEN T=190:PRINT"E#";:GOSUB 160
80 IF A$="Y" THEN T=179:PRINT"F#";:GOSUB 160
90 IF A$="U" THEN T=169:PRINT"F#";:GOSUB 160
100 IF A$="I" THEN T=159:PRINT"G#";:GOSUB 160
110 IF A$="O" THEN T=150:PRINT"G#";:GOSUB 160
120 IF A$="P" THEN T=142:PRINT"A#";:GOSUB 160
130 IF A$="@" THEN T=134:PRINT "A# ";:GOSUB 160
140 IF A$="[" THEN T=142:PRINT "A ";:GOSUB 160
150 GOTO 10
160 SOUND 1,T,30,15
170 RETURN
```

Make sure that the CAPS LOCK is switched on, then run the program. Now press any of the keys from Q to [ on the top row of letter keys. As you will hear, these have been programmed to play the notes of a scale. The name of the note being played also appears on the screen.

To play a note, just press its key once. Don't keep your finger on it as this causes the note to repeat automatically.

To finish the program, press key S.

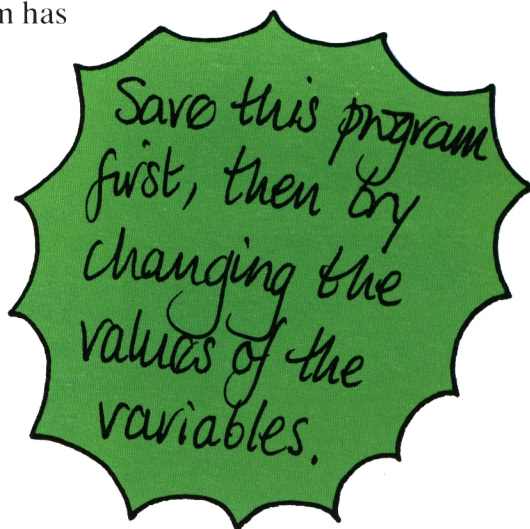




Here is another program for you to type in and save. Can you guess what it does?

To clear the screen, just press any key after the program has run.

```
10 MODE 0
20 PAPER 3:CLS
30 X=60:Y=390:B=10:GOSUB 130
40 X=250:GOSUB 130
50 X=450:Y=75:GOSUB 130
60 X=450:E=100
70 FOR T=390 TO 100 STEP-1
80 PLOT X,T:DRAW X+E,T
90 X=X+0.125:E=E-0.25
100 NEXT T
110 IF INKEY$="" THEN 110
120 MODE 1:PAPER 0:CLS:END
130 FOR A=X TO X+100
140 PLOT A,B,1:DRAW A,Y
150 NEXT
160 RETURN
```





\*MORFAX\*

The word INPUT is very useful. It allows you to change the information used by a program without changing the program listing.

Information fed into a program while it is running is sometimes also called INPUT.

# Putting in Information

You don't need to be a computer programmer to be able to use a computer. People use computers all the time. They may never see the programs that they use, nor do they need to do so. When you play a computer game you probably won't see the program for it and even if you do, you don't have to understand it. This is because computers let you feed information into them directly.

## INPUT

Using the BASIC word INPUT you can put information directly into your Amstrad CPC464's memory. This saves you the trouble of listing and changing your programs all the time. Let's look at how to use INPUT. Type in:

```
10 CLS
20 PRINT "GIVE ME A NUMBER"
30 INPUT A
40 PRINT "YOU TYPED IN";A
```

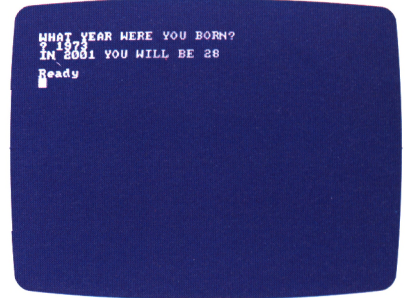
Now run the program. You will see the question printed and, on the next line, a question mark. This is the Amstrad CPC464's way of telling you that it wants you to put in or INPUT information. Type in your answer, press ENTER and the computer will tell you what number you typed in.

Once your answer is stored in the computer's memory as the variable A, the computer can use that information. Run the program a few times, and put in a different answer each time to see what happens.



All sorts of information can be INPUT and used in your programs. This next program is called '2001'. Type it in and run it to see why:

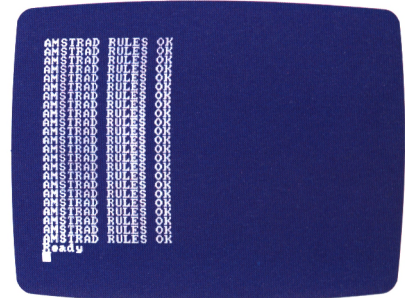
```
10 CLS
20 PRINT "WHAT YEAR WERE YOU BORN?"
30 INPUT YEAR
40 PRINT "IN 2001 YOU WILL BE ";2001-YEAR
50 PRINT
```



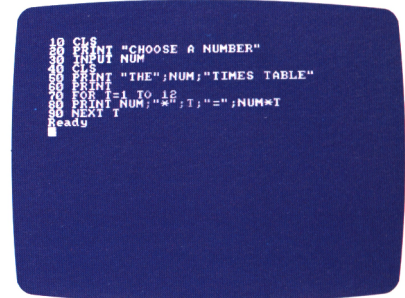
## Changing loops

You can use INPUT to control the size of a loop. This means you can ask your Amstrad CPC464 to do something however many times you choose. Type in this next program to see how this works:

```
10 CLS
20 PRINT "HOW MANY PRINTINGS"
30 INPUT N
40 FOR X=1 TO N
50 PRINT "AMSTRAD RULES OK"
60 NEXT X
```

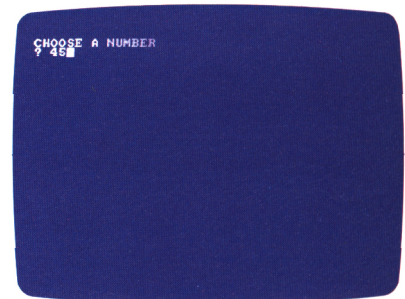


Now run the program. Don't forget to press ENTER after you've typed in your number. Your message will be printed as many times as you choose. Why not change the message at line 50?

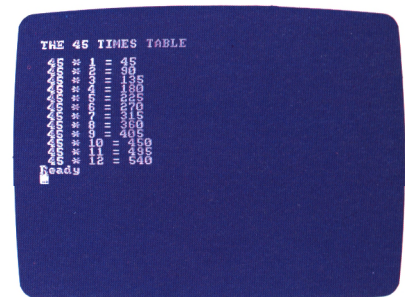


This next program is a more serious use of INPUT. You ask your Amstrad CPC464 to print out the multiplication table of any number you choose. The number is stored as the variable NUM. Type in:

```
10 CLS
20 PRINT "CHOOSE A NUMBER"
30 INPUT NUM
40 CLS
50 PRINT "THE";NUM;"TIMES TABLE"
60 PRINT
70 FOR T=1 TO 12
80 PRINT NUM;"*";T;"=";NUM*T
90 NEXT T
```



Run the program, type in your number and then press ENTER. The program will print out the multiplication table for your number.





## Putting in words

You can also INPUT words and strings. This is almost exactly the same as inputting numbers, but you have to remember to use the \$ sign after the variable name. Try this now:

```
10 CLS
20 PRINT "HELLO, WHO ARE YOU ?"
30 INPUT NAME$
40 CLS
50 PRINT "HELLO ";NAME$
60 PRINT
```

Run the program, type in your name and press ENTER. Of course your Amstrad CPC464 hasn't really recognized you. If you were to type in any name, even numbers, the program would still say Hello.

These lines could be put at the start of any of the programs you write, to make them more USER FRIENDLY.



```
10 CLS
20 INPUT "WHO ARE YOU ";NAME$
30 CLS
40 PRINT "HELLO ";NAME$
50 PRINT
60 PRINT
```

```
HELLO LYNN
Ready
```

REMEMBER  
to press  
ENTER  
after typing  
in your  
answers.

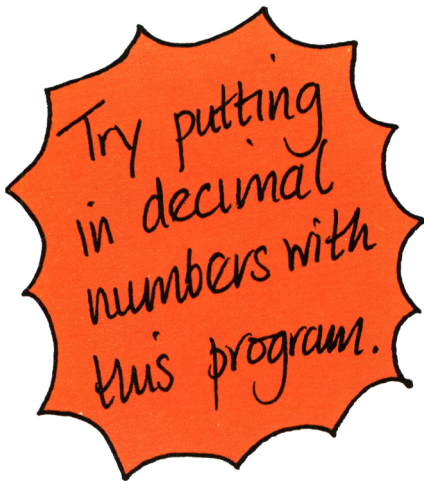
### \*MORFAX\*

You can use INPUT on its own without PRINT. To do this, type in:

```
10 CLS
20 INPUT "WHO ARE YOU ";NAME$
30 CLS
40 PRINT "HELLO ";NAME$
50 PRINT
```

This saves memory space and is easier to type in.





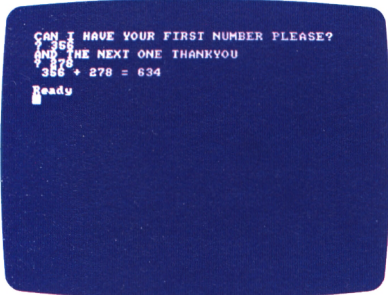
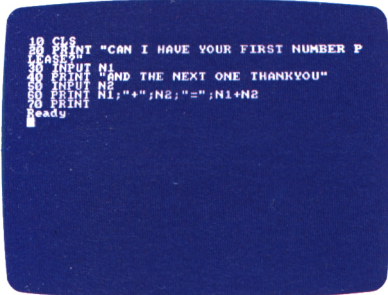
# Calculator programs

Thanks to INPUT you can write a program to do your addition sums. When you run it, all you need to do is type in the numbers you want added up.

Try to get the spacing right so that your screen display looks neat. Type in:

```
10 CLS
20 PRINT "CAN I HAVE YOUR FIRST NUMBER
PLEASE ?"
30 INPUT N1
40 PRINT "AND THE NEXT ONE THANK YOU."
50 INPUT N2
60 PRINT N1;"+";N2;"=";N1+N2
70 PRINT
```

Run the program now. Press ENTER after you've given each number. Try it several times, then have a go at writing a program that adds three numbers together.



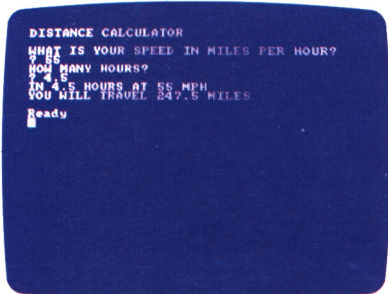
## \*MORFAX\*

In this last program you don't really need lines 20 or 40, but they do make the program easier for people to use and understand. You should always aim to make your programs USER FRIENDLY.

## Speedometer

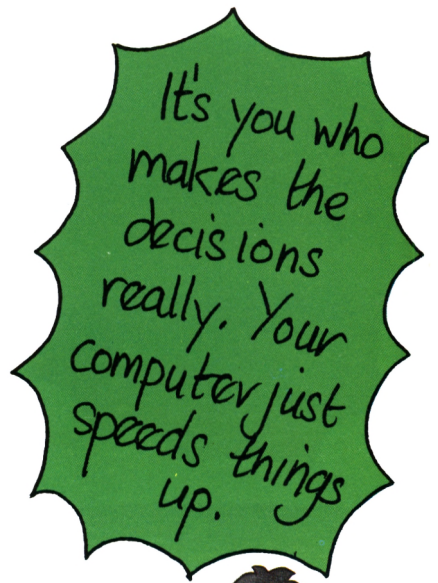
This next program uses INPUT with multiplication. It tells you how far you would go in a certain time, at a particular speed. Type in:

```
10 CLS
20 PRINT "DISTANCE CALCULATOR":PRINT
30 PRINT "WHAT IS YOUR SPEED IN MILES PER
HOUR ?"
40 INPUT SP
50 PRINT "HOW MANY HOURS ?"
60 INPUT NUM
70 PRINT "IN";NUM;"HOURS AT";SP;"MPH"
80 PRINT "YOU WILL TRAVEL";SP*NUM;
"MILES"
90 PRINT
```





# Making Decisions



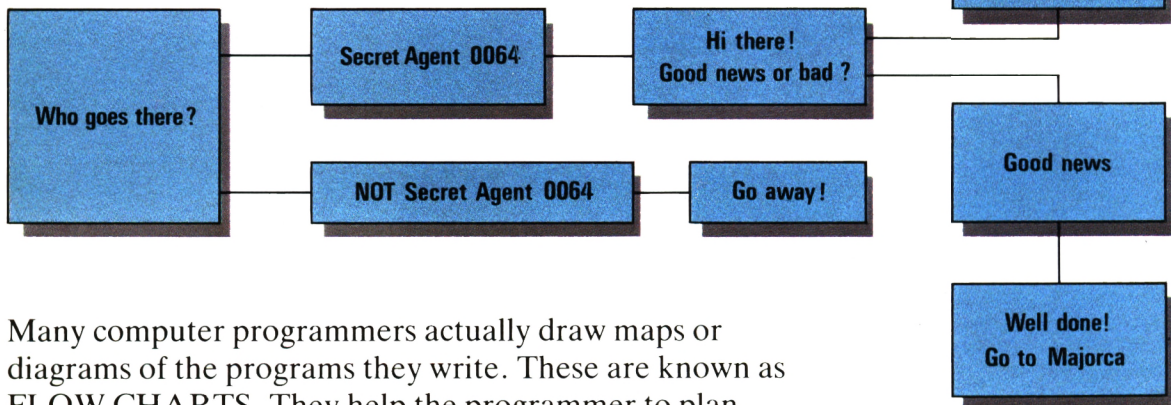
Computers can help us to make decisions and choices. They can be programmed to find and select information, and to respond to that information. Most of the important work done by computers involves making choices.

Can you think of the kinds of decision computers help to make? Have you renewed your motor tax? . . . Is there enough money in the bank to pay the bills? . . . Are there any seats on that holiday flight to Bermuda?

When a computer makes a decision, it checks through the information stored in its memory. IF it finds the number or words it is looking for, THEN it will carry out one set of instructions. IF it doesn't, THEN it will carry out another set of instructions.



Because of this, many programs contain not just one but many sets of instructions. They become rather like a road map showing all sorts of different routes and destinations. Depending on the answers and responses you give to the computer, you may be given a number of different answers.



Many computer programmers actually draw maps or diagrams of the programs they write. These are known as FLOW CHARTS. They help the programmer to plan programs.



## \*MORFAX\*

What makes computers so useful is the speed with which they can sort through information. Today, large mainframe computers are used by many organisations to store their records. Instead of having to spend hours searching through shelves full of files, facts and figures can be easily collected and checked using the power of computers.

## IF . . . THEN

To program your Amstrad CPC464 to make decisions, you use the BASIC words IF and THEN. Let's look at a simple decision program.

We shall program your Amstrad CPC464 to recognize a name. IF it recognizes that name THEN it will say Hello. Type in the first part of the program:

```
10 CLS
20 INPUT "WHO GOES THERE ";NAME$
30 IF NAME$="AMSTRAD" THEN PRINT "HELLO "
   ;NAME$
```

Run the program. Type in AMSTRAD (in capitals) as your answer, and the computer will say hello to you.

## IF NOT . . .

Now let's tell your Amstrad CPC464 what to do if your name is not AMSTRAD. This time you use the BASIC word NOT after the word IF. Type in:

```
40 IF NOT NAME$="AMSTRAD" THEN PRINT
   "GO AWAY ";NAME$
```

Now run the program. First pretend that your name is AMSTRAD. Next time you run it, type in your own name. As you will see, the program is not quite so friendly when you do this!

What happens is that the program decides which answer to give by looking at your reply and comparing it with the answer AMSTRAD.

```
10 CLS
20 INPUT "WHO GOES THERE ";NAME$
30 IF NAME$="AMSTRAD" THEN PRINT "HELLO "
   ;NAME$
Ready
```

```
WHO GOES THERE ? AMSTRAD
HELLO AMSTRAD
Ready
```

## \*MORFAX\*

Your Amstrad CPC464 treats capital letters as being different from lower case (little) letters. It will not regard 'Amstrad' as being the same as 'AMSTRAD'.

```
10 CLS
20 INPUT "WHO GOES THERE ";NAME$
30 IF NAME$="AMSTRAD" THEN PRINT "HELLO "
   ;NAME$
40 IF NOT NAME$="AMSTRAD" THEN PRINT "GO
   AWAY ";NAME$
Ready
```

```
WHO GOES THERE ? KEVIN
GO AWAY KEVIN
Ready
```



# Quiz programs

You can use IF and THEN to test your friends' general knowledge. This next program is a simple quiz program. It tests your knowledge of capital cities and uses IF and THEN in a very simple way. Type in:

```
10 REM * CAPITAL CITIES QUIZ *
20 BORDER 24:PAPER 9:PEN 0:CLS
30 PRINT "CAPITAL CITIES QUIZ"
40 PRINT
50 INPUT "WHAT IS THE CAPITAL OF FRANCE";CITY$
60 IF CITY$="PARIS" THEN PRINT "CORRECT"
70 IF NOT CITY$="PARIS" THEN PRINT "NO, WRONG."
80 INPUT "WHAT IS THE CAPITAL OF SPAIN";CITY$
90 IF CITY$="MADRID" THEN PRINT "CORRECT"
100 IF NOT CITY$="MADRID" THEN PRINT "NO,WRONG."
110 INPUT "WHAT IS THE CAPITAL OF ITALY";CITY$
120 IF CITY$="ROME" THEN PRINT "CORRECT"
130 IF NOT CITY$="ROME" THEN PRINT "NO,WRONG"
140 IF INKEY$="" THEN 140
150 BORDER 1:PAPER 0:PEN 9:CLS
```

REMEMBER  
the quotation  
marks when  
you use IF  
with words.

## CAPITAL CITIES QUIZ

```
WHAT IS THE CAPITAL OF FRANCE? PARIS
CORRECT
WHAT IS THE CAPITAL OF SPAIN? MADRID
CORRECT
WHAT IS THE CAPITAL OF ITALY? MILAN
NO,WRONG
```

## \*MORFAX\*

The word REM at line 10 stands for REMARK. The rest of the line is a description of the program. This line doesn't affect the program but acts like a title, so that when you look at the listing you know what the program is about. Using REM like this helps to make your listings clearer.

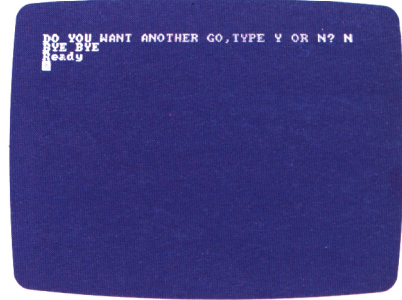
Why not add some more lines to this program, or write a quiz of your own? Your quiz could be about all sorts of things, from football to pop music.



## Another go?

Using IF and THEN with the BASIC word GOTO, you can program your computer to let you have another go at your quiz. The next few lines (provided you alter the line numbers) can be put at the end of any of your programs. They allow you to have another go at running the program. Add them to the program you have just written. Don't forget to press ENTER when you've given your answer.

```
160 INPUT "DO YOU WANT ANOTHER GO, TYPE  
YES OR NO ";ANS$  
170 IF ANS$="YES" THEN GOTO 10  
180 IF NOT ANS$="YES" THEN PRINT "BYE BYE"
```

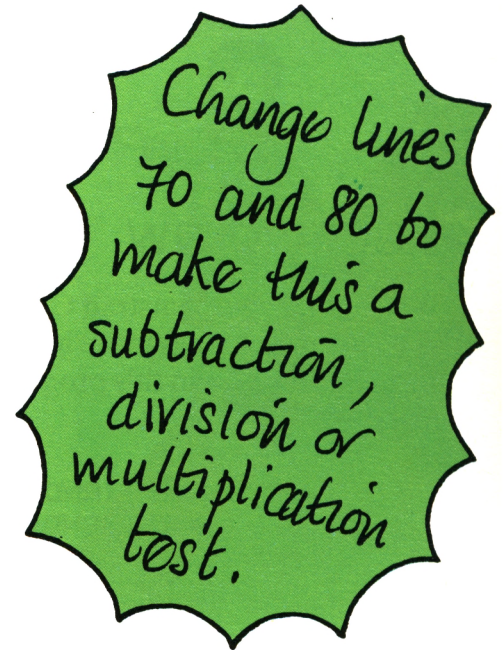


## Test your maths

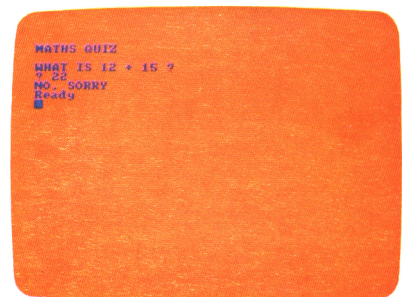
You can use IF . . . THEN with numbers too. The last few pages in this book will show you how you can write a program using many of the BASIC words and instructions we've looked at so far. It will also introduce some useful new signs.

The next program tests your maths skills! It asks you to add two numbers together. It then tells you whether or not you have given the right answer. Type in:

```
10 REM * MATHS QUIZ *  
20 BORDER 6:PAPER 3:PEN 0:CLS  
30 PRINT "MATHS QUIZ"  
40 PRINT  
50 LET N1=12  
60 LET N2=15  
70 LET RIGHT=N1+N2  
80 PRINT "WHAT IS";N1;"+";N2;"?"  
90 INPUT ANS  
100 IF ANS=RIGHT THEN PRINT "WELL DONE,  
CORRECT"  
110 IF NOT ANS=RIGHT THEN PRINT "SORRY,  
WRONG"
```



Run the program now, typing in the wrong and the right answer. Can you see how the right answer has been stored as a variable, at line 70? When you want to make the quiz more difficult, just change the numbers stored as N1 and N2.



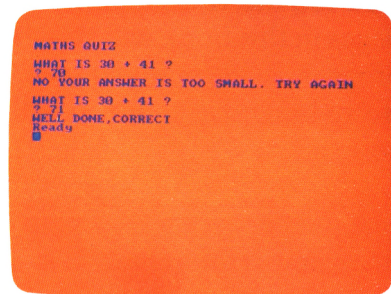
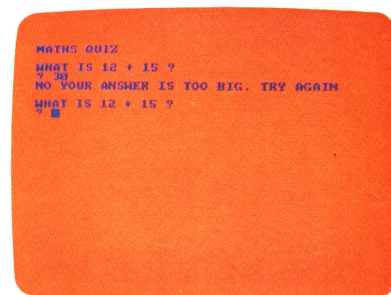


## Too big or too small?

You can change this program so that it gives you a little more help. It can tell you whether your wrong answer is too big or too small. To do this you use the symbols < and > (bottom row of the keyboard, with SHIFT). The > sign stands for greater or bigger than. The < sign stands for less or smaller than.

The word GOTO helps by letting you 'jump' to another line or part of a program. As you make the questions in your programs more difficult, you will probably need more than one go to get the right answer. Here, GOTO sends your program back to line 90. Type in these lines:

```
110 IF ANS>RIGHT THEN PRINT "NO,YOUR  
ANSWER IS TOO BIG. TRY AGAIN":PRINT:GOTO 90  
120 IF ANS<RIGHT THEN PRINT "NO,YOUR  
ANSWER IS TOO SMALL. TRY AGAIN":PRINT:  
GOTO 90
```



## Lucky draw

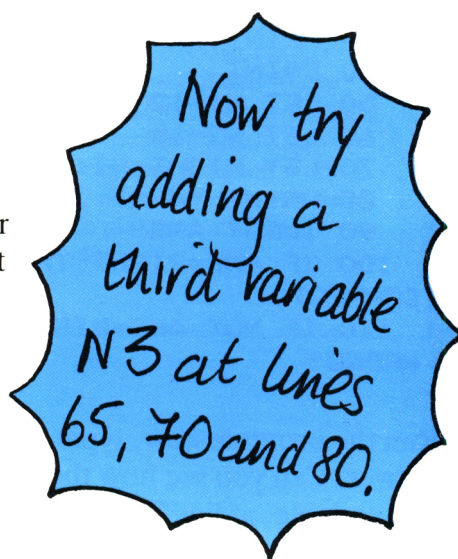
Why not let your Amstrad CPC464 choose the numbers for you and make things more interesting? You can use RND with the numbers in this program. Change lines 50 and 60 like this:

```
50 LET N1=INT(RND*50)+10  
60 LET N2=INT(RND*50)+10
```

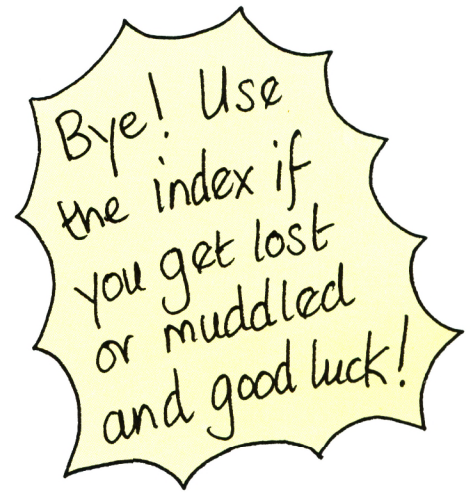
Each time you run it now, you'll get a different set of numbers and a different sum to do. If you want to use bigger numbers, decide on the range you want. Use that number at lines 50 and 60, and away you go. To make it even harder, remove the word INT from both lines!

You can also give yourself another go at the quiz after you get the answer right. Type in:

```
130 INPUT "ANOTHER GO, Y OR N";ANS  
140 IF ANS="Y" THEN GOTO 10  
150 BORDER 1:PAPER 0:PEN 1:CLS
```







Well, that's all for now. I hope you've enjoyed this book and that it has given you lots of ideas. Don't stop now though. You've only just started! Be adventurous and start to write your own programs. Be prepared for hard work, a few mistakes and many hours of fun!





# Index

- Addition 12, 61
- AUTO 24
- 
- BASIC 6, 7, 10, 27, 28
- Big numbers 13
- Bit 28
- BORDER 21
- Boxes 45, 46
- Break 9
- Brightness 6
- Byte 28
- 
- Calculator programs 57
- Capitals 9, 31, 59
- CAPS LOCK 9
- Cassette recorder 5, 7
- Chips 7, 27, 28
- Circles 47
- Clearing the screen (CLS) 9, 17, 37
- Colour 21, 22, 38, 39, 44
- Comma 13
- Computer, care of 4
  - languages 28
  - setting up 6
  - system 5
- Co-ordinates 42
- COPY 33
- CPU chip 27
- CTRL 9
- Cursor 8, 33
- 
- Decimal points 13
- DEG 47
- DEL 8
- Division 12, 13
- Dollar sign 32
- DRAW 44
- Drawing 42–48
- 
- EDIT 34
- ENTER 9, 11, 37
- ESC 9, 36
- 
- FOR . . . NEXT 34, 35
- 
- GOTO 61, 62
- Graphics 46
- 
- Hardware 5
- 
- IF NOT 59
- IF . . . THEN 59
- INK 39
- INPUT 54–57
- INT 38, 62
- 
- KEY 37
- Keyboard 8, 9
- K (Kilobytes) 28
- 
- LET 29
- Line numbers 16–18
- Listing 17, 37, 43
- Loading programs 49, 51, 52
- Longer variable names 30
- Loops 34–37, 41
- 
- Machine code 27, 28
- Memory 7, 18, 28, 29
- Modes 20–22, 43
- Monitor 5, 6
- Multiplication 12–14, 36
- Multi-statement lines 45
- Music 40, 52
- 
- NEW 18
- Numeric keypad 9, 37
- 
- PAPER 21, 22
- PEN 21, 22
- Pixels 42
- PLOT 43
- Power switches 6
- PRINT 11, 13, 14, 19, 37, 56
- 
- Programs 7, 14, 16, 58
  - to try 23–26, 52, 53
- Program lines, changing 18
  - editing 33
  - removing 18
  - line numbers 16–18, 24
- 
- Quiz programs 60
- Quotation marks 14
- 
- RAM 28
- Ready 6, 11, 26, 37, 51
- REM 60
- RENUM 18
- Report messages 11
- Resetting your computer 9
- RND 38, 39, 41, 43
- ROM 27, 28
- Rubbing out 8
- RUN 16
- 
- Saving programs 49, 50
- Screen display 19
- Scrolling 16, 17, 36
- Semi-colon 14
- Setting up 6
- SHIFT keys 9
- Software 7, 10, 51
- Sound chip 27, 28
- Sound effects 26, 40, 41
- Spacing 8, 11, 21, 31
- STEP 35
- Strings 14, 32
- Subtraction 12
- Syntax error 10, 31, 32
- 
- Tables 36, 55
- 
- User friendliness 31, 56, 57
- 
- Variables 29–32
- VDU screen 5
- 
- > and < 62







**BEGINNERS' MICRO GUIDES**

# **Amstrad CPC464**



**So, you've got your computer?  
What do you do next?**

**Don't worry, this book will tell you what you need to know, from how to set it up and switch it on, to solving problems, playing games and even making music.**

**An ideal choice for child, parent or teacher, this Beginners' Micro Guide explains how to use the computer and gives you sample programs which can be adapted by you to suit your needs. The uncomplicated words and diagrams will encourage you to learn by experiment and make the most of your micro.**



# JOHNATHAN GLISMAN'S FAVORITE POSTCARDS FROM GRANDMA